The 6th International Summer Workshop on Multimodal Interfaces

eNTERFACE '10

PROCEEDINGS eNTERFACE'10

Summer Workshop on Multimodal Interfaces

July 12 - August 6, 2010
University of Amsterdam - The Netherlands

Editors:
Albert Ali Salah, Theo Gevers

Proceedings eNTERFACE'10

6th International Summer Workshop
on Multimodal Interfaces

July 12 – August 6, 2010
University of Amsterdam, The Netherlands

Editors:
Albert Ali Salah, Theo Gevers

Printed by Xerox Nederlands BV, the Netherlands

Edited by
Albert Ali Salah
University of Amsterdam
Informatics Institute
1098 XG Amsterdam, The Netherlands
E-mail: a.a.salah@uva.nl

Theo Gevers
University of Amsterdam
Informatics Institute
1098 XG Amsterdam, The Netherlands
E-mail: th.gevers@uva.nl

ISBN 978-90-5776-213-0

Cover design
Hamdi Dibeklioğlu & Alkım Almıla Akdağ Salah

http://enterface10.science.uva.nl/
http://isla.science.uva.nl/

UNIVERSITEIT VAN AMSTERDAM

Organization

Co-chairs

Albert Ali Salah	University of Amsterdam
Theo Gevers	University of Amsterdam

Scientific Committee

Lale Akarun	Boğaziçi University
Antonio Camurri	University of Genoa
Cristophe d'Alessandro	CNRS-LIMSI
Thierry Dutoit	Faculté Polytechnique de Mons
Theo Gevers	University of Amsterdam
Ben Kröse	University of Amsterdam
Maurizio Mancini	University of Genoa
Panos Markopoulos	Eindhoven University of Technology
Ferran Marqués	Universitat Politécnica de Catalunya
Ramon Morros	Universitat Politécnica de Catalunya
Anton Nijholt	University of Twente
Igor Pandzic	Zagreb University
Catherine Pelachaud	CNRS-TELECOM Paris-Tech
Albert Ali Salah	University of Amsterdam
Bülent Sankur	Boğaziçi University
Ben Schouten	Eindhoven University of Technology
Björn Schuller	Technical University of Munich
Nicu Sebe	University of Trento
Alessandro Vinciarelli	University of Glasgow
Gualtiero Volpe	University of Genoa

Preface

One month seems to be quite long for a workshop, but for the participants of eNTERFACE, the workshop ended just when we started to get to know each other. It was a great pleasure to organize and host the eNTERFACE, and to work with so many bright people with common scientific interests.

Let me briefly describe the eNTERFACE to the initiate... This series of workshops aim at establishing a tradition of collaborative research by gathering, in a single place, teams of students and senior researchers of multimodal man-machine interfaces, to work on a pre-specified list of challenges, for four complete weeks. In this respect, it is an innovative and intensive collaboration scheme, designed to allow researchers to integrate their software tools, deploy demonstrators, collect novel databases, and work side by side with a host of experts. The eNTERFACE was born in 2005 as the yearly workshop of the SIMILAR FP6 Network of Excellence on multimodal human-computer interaction. After the completion of SIMILAR, the workshop continued to attract wide interest under the aegis of the OpenInterface Foundation. It was organized by Faculté Polytechnique de Mons in 2005, University of Zagreb in 2006, Boğaziçi University in 2007, CNRS-LIMSI in 2008, and University of Genoa in 2009. The 6th eNTERFACE'10 Workshop of Multimodal User Interfaces (to give its full name) was hosted by the Intelligent Systems Lab Amsterdam (ISLA-ISIS) of the University of Amsterdam during 12 July-6 August 2010. This volume collects the project reports from each of the seven projects completed in the workshop.

The eNTERFACE'10 in Amsterdam brought together 65 researchers from 18 countries, who not only worked together during this intense month, but also explored the city, and had fun together. Early in the first week we had a wine & cheese poster session to learn about each others research, and to socialize. It was a very successful session, to the extent that the security had to ask us to vacate the building at the end, as they were closing it for the night. We thank Dr. Catherine Pelachaud for the idea.

Apart from the projects, we had seven invited plenaries and three tutorials, enabled by the generous support of SenterNovem IOP-MMI, the SSPNet FP7 Network of Excellence on Social Signal Processing, ERASMUS and EURASIP. All the material accumulated during the workshop (including code, data, reports, presentations and tutorials), as well as material from all the previous editions of the workshop, can be found online, at http://www.enterface.net/.

I would like to thank our distinguished speakers (in order of appearance) Dr. Marc Schröder (DFKI), Dr. Hamid Aghajan (Stanford University), Dr. Anton Nijholt (University of Twente), Esther Polak (independent artist), Dr. Ben Schouten (Eindhoven University of Technology), Dr. Leon Rothkrantz (Delft University of Technology), Dr. Hakan Erdoğan (Sabancı University), and Dr. Josef Kittler (University of Surrey). I am grateful for the support of our excellent scientific committee, in particular Dr. Dirk Heylen (University of Twente) for organizing the SSPNet talks and tutorials.

The organization of the eNTERFACE is a lengthy process. It would have been impossible to organize it without the support of the University of Amsterdam, and ISLA-ISIS. I would especially like to thank my co-chair Dr. Theo Gevers for his support, and Dr. Thierry Dutoit for his encouragement. A big "thank you!" goes to our department secretary Virginie Mes, and to Geert Olthof, for their help in organizational and financial matters. Finally, I would like to thank my wife Almıla Akdağ Salah for designing the website, and Hamdi Dibeklioğlu for his many contributions.

Albert Ali Salah, co-chair
Amsterdam, 2010.

The wine & cheese poster session.

eNTERFACE participants during a presentation.

Program

Mon. July 12
: General opening meeting.
Project presentations.

Tue. July 13
: Teams gathering and installation.
Wine & cheese poster session.

Wed. July 14
: SSPNet Plenary: **Marc Schröder** (DFKI) – "OpenMary Text-to-Speech," followed by tutorial.

Thu. July 15
: SSPNet Plenary: **Marc Schröder** (DFKI) – "Building Emotion-oriented Real-time Interactive Systems with the SEMAINE API," followed by tutorial.

Mon. July 19
: Invited talk: **Hamid Aghajan** (Stanford University) – "Ambient Intelligence: From Sensor Networks to Smart Environments and Social Networks"

Wed. July 21
: Invited talk: **Anton Nijholt** (University of Twente) – "People as Content"

Mon. July 26
: Invited talk: **Esther Polak** (independent artist) & **Ben Schouten** (Eindhoven University of Technology) – "Interactive Information Visualization"

Tue. July 27
: Midterm presentations.
Intermediate reports on teams achievements.

Wed. July 28
: Invited talk: **Leon Rothkrantz** (Delft University of Technology) – "Surveillance by Multimodal Camera Systems"

Mon. Aug 2 – Thu, Aug 5
: ERASMUS Tutorial: **Hakan Erdoğan** (Sabancı University) – "Structured Learning Approaches for Sequence Labeling"

Wed. Aug 4
: EURASIP Plenary: **Josef Kittler** (University of Surrey) – "Information Fusion in Content-based Retrieval from Multimedia Databases"

Fri. Aug 6
: Final project presentations and concluding remarks.

Table of Contents

CoMediAnnotate: towards more usable multimedia content annotation by adapting the user interface

Christian Frisson [1,2,n], Sema Alaçam [3], Emirhan Coşkun [3], Dominik Ertl [4], Ceren Kayalar [5], Lionel Lawson [1], Florian Lingenfelser [6], Johannes Wagner [6]

[1] Communications and Remote Sensing (TELE) Lab, Université catholique de Louvain (UCLouvain), Belgium;
[2] Circuit Theory and Signal Processing (TCTS) Lab, Université de Mons (UMons), Belgium;
[n] numediart Research Program on Digital Art Technologies;
[3] Architectural Design Computing, Institute of Science and Technology, Istanbul Technical University (ITU), Turkey;
[4] Institute of Computer Technology, Vienna University of Technology, Vienna, Austria;
[5] Computer Graphics Lab (CGLab), Sabancı University, Istanbul, Turkey;
[6] Lehrstuhl für Multimedia-Konzepte und Anwendungen (MM), Universität Augsburg, Germany

***Abstract*— This project aims at improving the user experience regarding multimedia content annotation. We evaluated and compared current timeline-based annotation tools, so as to elicit user requirements. We address two issues: 1) adapting the user interface, by supporting more input modalities through a rapid prototyping tool and by offering alternative visualization techniques of temporal signals; and 2) covering more steps of the annotation workflow besides the task of annotation itself: notably recording multimodal signals.**

We developed input devices components for the OpenInterface (OI) platform for rapid prototyping of multimodal interfaces: multitouch screen, jog wheels and pen-based solutions. We modified an annotation tool created with the Smart Sensor Integration (SSI) toolkit and componentized it in OI so as to bind its controls to different input devices. We produced mockups sketches towards a new design of an improved user interface for multimedia content annotation, and started developing a rough prototype using the Processing Development Environment.

Our solution allows to produce several prototypes by varying the interaction pipeline: changing input modalities and using either the initial GUI of the annotation tool, or the newly-designed one. We target usability testing to validate our solution and determine which input modalities combination best suits given use cases.

***Index Terms*— Multimodal annotation, rapid prototyping, information visualization, gestural interaction**

I. Introduction

This project attempts to provide a tentative toolbox aimed at improving the user interface of current tools for multimedia content annotation. More precisely, this project consists in combining efforts gathered in fields such as rapid prototyping, information visualization, gestural interaction; by adding all the necessary and remaining components to a rapid prototyping tool that allows to visually program the application workflow, in order to the refine the user experience, first of one chosen annotation tool. This toolbox is a first milestone in our research, a necessary step to undertake usability tests on specific scenarios and use cases after this workshop.

This report is structured as follows. In Section II we define the context and scope of the project, i.e. "*multimedia content*" (Section II-A) "*annotation*" (Section II-B) and list possible use cases (Section II-C) and testbeds (Section II-D). In Section III, we summarize the current problems of timeline-based multimedia content annotation tools, based on previous comparisons (Section III-A) and on an evaluation we undertook during the workshop (Section III-B), then we explain why we chose to adapt the SSI annotation tool (Section III-C). In Section IV, we describe the method we opted for: through a user-centered approach (Section IV-A), we restricted our design to two modalities (Section IV-B): visualization (Section IV-B.1) and gestural input (Section IV-B.2), among other possible modalities (Section IV-B.3); we thus used a rapid prototyping (Section IV-C) visual programming tool (Section IV-C.1) for the user interface (Section IV-C.2), the OpenInterface platform (Section IV-C.2.b), and a rapid prototyping tool for visualization (Section IV-C.3), the Processing Development Environment (Section IV-C.3.b). In Section V, we summarize our results: we proposed a new tentative design of an improved user interface (Section V-A), illustrated with mockups (Section V-A.2) and an early prototype (Section V-A.3); and we developed components for the OpenInterface platform (Section V-B) for gestural input modalities (Section V-B.1), control of the SSI annotation tool (Section V-B.2). In Section VI, we underline our future works: a more robust prototype integrated into the MediaCycle framework for multimedia content navigation by similarity (Section VI-A) and subsequent usability tests to validate our designs (Section VI-B). Finally, we conclude in Section VII.

II. Context: annotation of multimedia content

A. What do we mean by "multimedia"

"Multimedia data" commonly refers to content (audio, images, video, text...) recorded by sensors and manipulated by all sorts of end-users. In contrast, the term "multimodal data" describes signals that act as ways of communication between humans and machines. Multimodal data can be considered as of a subset of "multimedia data", since the first are produced by human beings. Multimedia data thus broaches a wider range of content (natural phenomena, objects, etc...). Annotation tools help analyzing multimedia data, but also make use of multimodal signals within their user interface.

B. What do we mean by "annotation"

The following questions illustrate the issues we faced while understanding each others on a generic definition of the term "annotation":

- Who is doing it? Human(s) and/or machine(s)?:
  - automatic annotation consists in extracting metadata using signal processing algorithms with no (or limited) parameter tweaking required from the user;
  - "manual" annotation is performed by humans adding metadata to data using various user interaction techniques;
  - semi-automatic annotation combines both approaches, sequenced in time. For instance: once data is loaded in the annotation tool, feature extraction algorithms run in the

background on a subset, the user is then asked to correct these automated annotation, then a process propagates the corrections to the whole dataset.

- In case of humans, what about standard versus expert users? Is it being performed collaboratively by multiple users?
- What kind of data is annotated? "Multimedia content" and/or "multimodal signals"?
- When is it performed? Online and/or offline?
- For what purpose? Which use cases, scenarii?

Semiotically, from the user perspective and viewpoint, two types of annotation can be discriminated:

- *semantic*: words, concepts... that can be assorted in domain ontologies,
- *graphic*: baselines, peaks... with a tight relation to the gestural input required to produce them

Additionally, Kipp proposes a spatio-temporal taxonomy in [29]: *shape* (or geometric representation), *number* (of occurrences), *order* (chronological or not), *rigidity* and *interpolation* (discrete, linear or spline).

We opted for the following definition: annotations consist in "*adding metadata to data in order to extract information*", that is contradictory with [42] which confronts "*annotation*" and "*metadata*", the first term considered time-dependent by the author while the second isn't.

C. Possible use cases

We had in mind to propose a toolbox with which the user can adapt the annotation tool to his/her needs, instead of having to use a different tool for each domain of use, for instance: corpora archival, multimedia library sorting, sensor recordings analysis, etc.. There are numerous possible uses of multimedia content annotation, here follows a subset applied to multimedia arts:

- annotation of motion capture [23], for instance with online errors notification while recording for offline reconstruction of missing data;
- analysis of dancers' performances [56] requiring diverse types of sensors, training mappings of gesture-based dancers interfaces using performances recordings [16, 19];
- preparation of material for live cinema performances [37];
- multimedia archival and restoration [49]...

D. Possible testbeds

We tried to adapt the project scope to fit it better to some MSc/PhD participants topics, by considering two more testbeds besides timeline-based annotation tools.

1) Timeline-based Annotation Tools: focus on the analysis of temporal signals or time-series and offer a great challenge regarding handling time for navigation and annotation purposes. It has to be noted that most participants already had some experience with multimedia edition tools, requiring similar navigation methods and offering a subset of the variety of possible annotations.

2) Multimedia Collection Managers: such as iTunes for music libraries, sharing design questions with Emirhan's MSc (2D visualization and representation of massive datasets, in his case in the context of social networks) and Christian's PhD (similar, applied to multimedia content). We discarded this testbed because we haven't found any already-existing opensource tool that would offer flexible annotation further than basic metadata management (ID3 tags for music, movie "credits" information, etc...) for audio and video media types (however, we found some for image or text).

3) Panaromic Image based 3D Viewers and VR World Viewers: such as Google Earth, HDView Gigapixel Panoramas [32] and Photosynth [55], interesting particularly regarding Ceren's PhD work [27]. We discarded this testbed because developing a simple 3D viewer with annotation support or even integrating navigation and annotation through Google Earth API would have taken too much time, leaving not much time to deal with real research issues (for example: occlusion-free tag 3D position considering a variable user viewpoint).

III. Timeline-based Multimedia Content Annotation Tools: From Problems, towards User Requirements

A. Summary of current problems

Plenty of pre-existing works compared annotation tools and elicited emerging requirements, for instance throughout the last decade [6, 8, 12, 48, 50]. Based on these observations and readings, we summarize the following issues regarding how annotation tools could be improved (checked boxes emphasizing the ones we planned to address throughout the workshop):

- ☐ multimedia: better file formats support [6, 50], time-based media other than audio and video [6];
- ☐ scale: number and/or length of media elements in the database;
- ☐ reusability: toolboxes/frameworks rather than isolated tools specific to a given context of use [12, 48], portability over multiple operating systems [8, 50];
- ☐ accessibility: client/server applications rather than desktop applications working with local media databases;
- ☑ interactivity: a multimodal user interface could help enhance the pleasurability and efficiency of these tools that are generally WIMP-based [6, 12, 48], so as to provide a single used interface that allows;
  1) ☑ to monitor signal feeds while recording datasets,
  2) ☑ optionally to add annotations while recording,
  3) ☑ to edit or correct annotations;
  4) ☑ a more natural, usable, pleasurable user interface (pen and touch).
- ☑ workflow: supporting of the full annotation workflow [12, 18]:
  1) ☐ one administrator prepares (design of a template and choice of coders);
  2) ☑ several coders record;
  3) ☑ several coders annotate;
  4) ☐ the administrator analyses results (coder agreement...).

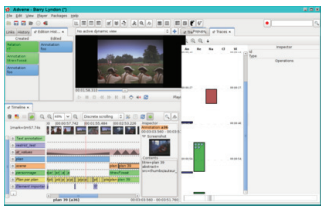
B. Evaluation and testing during eNTERFACE'10

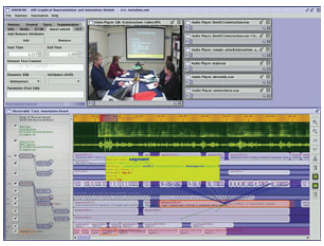
We tested 8 opensource or free tools, with screenshots in Fig 1, at least with one participant assigned to each (practically, two participants tested each), alphabetically: Advene [43, 1], AmiGram, Anvil [29, 28], ELAN [57], Lignes de Temps [41], On The Mark [64], Smart Sensor Integration (SSI) [61, 60] and VCode/VData [18, 58]; so as to better understand the concerns with a hands-on approach.

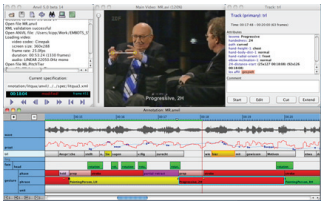
We produced detailed comparison in 3 tables that are available online on the eNTERFACE'10 wiki [1], focusing on:

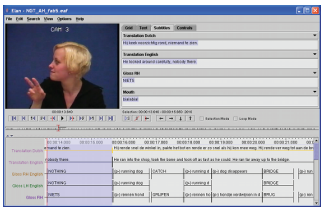
1) development criteria (quantitative): OS, licence, development languages, supported formats...;
2) context, usage (quantitative): media types, scope, field of use...;
3) eNTERFACE participants feedback (qualitative): subjective comments on usability and pleasurability raised by the participants while testing these tools.

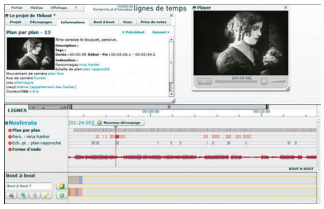
[1] http://enterface10.science.uva.nl/wiki/index.php/CoMediAnnotate:Framework:Annotation:Tools

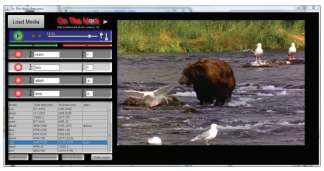

Advene [43, 1]


AmiGram [4]


Anvil [29, 28]


ELAN [57]


Lignes de Temps [41]


On The Mark[64]


SSI [61, 60]

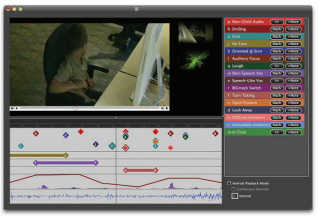

VCode/VData [18, 58]

Fig. 1
SCREENSHOTS OF OUR SELECTION AMONG THE AVAILABLE ANNOTATION TOOLS. IMAGE COPYRIGHTS REMAIN WITH THEIR AUTHORS.

A first round of selection based on development considerations (operating system, development language and licenses) narrowed down the choice among 3 candidates out of the 8 tested: AmiGram, ELAN and SSI.

- implementation: in C++ or Java or C# or Python, supported by the rapid prototyping platform for multimodal interfaces we chose (as explained in Section IV-C.2.b);
- license: necessarily open-source so that we could modify the source code;
- compatibility: running on most operating systems possible, the common denominator operating system among participants being Windows.

C. Chosen tool for adaptation: Smart Sensor Integration (SSI)

1) Description: The SSI toolkit [61, 62] developed within the CALLAS EU project by two of the participants, Johannes and Florian, is a framework for multimodal signal processing in real-time. It allows the recording and processing of human generated signals in pipelines based on filter and feature extraction blocks. By connecting a pipeline with a classifier it becomes possible to set up an online recognition system. The training of a recognition model requires the collection of a sufficient number of samples. This is usually accomplished in two steps: 1) setting up an experiment to induce the desired user behavior, 2) review the recorded signals and add annotation to describe the observed behavior. For this purpose SSI offers a an annotation tool for multimedia signals. Signals recorded with SSI can be reviewed and annotated within this tool (see Fig. 2).

Depending on the length of the recordings (usually several hours) annotation can turn out to be an extremely time-consuming task. Currently the tool is controlled via simple mouse and keyboard commands. This is not always the fastest way and after some while of continuous use can become inconvenient for the user. Hence, the tool would greatly benefit from alternative ways of interaction, such as Nintendo's WiiRemote control or a gamepad.

2) Reasons for the choice:

- We are in close contact with its developers who participated to the project during the first week.
- The core is separated from the UI.
- The simple annotation GUI is lightweight, hence simple to understand, and easy to replace.
- The toolkit not only proposes a simple annotation tool, but also feature extraction algorithms for automatic annotation, and could bridge the gap between multimedia content and multimodal signals annotation. This is of interest for some participants like Dominik for future works around adaptive multimodal interfaces by training [51].
- The development languages are compatible with the chosen rapid prototyping platform (see Section IV-C.2.b).

IV. METHOD

A. User-centered approach

We opted for a user-centered approach [12] to conduct our research:

- in addition to gathering scientific documentation, we undertook a small contextual inquiry with eNTERFACE participants that had already had to use an annotation tool;
- before diving into software development, we cycled through and brainstormed on different design propositions using paper mockups;
- we produced a fast software and hardware prototype with off-the-shelf devices using rapid prototyping tools, as a first proof-of-concept, before rethinking the prototype with a more dedicated but slower to implement solution.

B. Two modalities of interest

Currently, we target standard experts (ie not "disabled" users such as blind people), yet such cases could be addressed since we are making use of a rapid prototyping tool for multimodal user interfaces. Ever since before computerized systems, two modalities were deeply rooted in the task of annotation: visualization and gestural input.

Fig. 2

IN SSI RECORDED SESSIONS ARE VISUALIZED TOGETHER WITH ANNOTATION TRACKS THAT DESCRIBE THE OBSERVED USER BEHAVIOR. THE SCREENSHOT SHOWS FOUR SIGNALS (TOP DOWN: EYEGAZE, HEAD TRACKING, AUDIO AND BLOOD VOLUME PULSE) AND TWO ANNOTATION TRACKS (HERE: THE TRANSCRIPTION OF THE DIALOG BETWEEN THE VIRTUAL CHARACTER AND THE USER). ON THE LEFT SIDE VIDEOS OF THE APPLICATION AND THE USER ARE DISPLAYED. SCREENSHOT FROM [62].

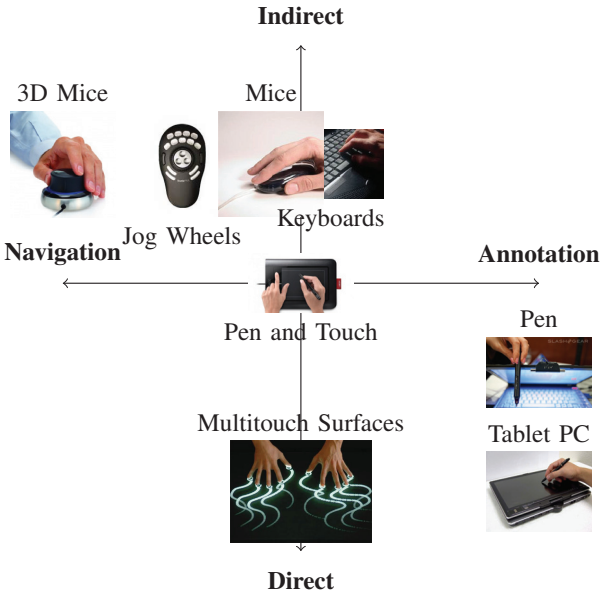


Fig. 3

A SELECTION OF DEVICES SORTED ON A 2-D SPACE, INDICATING: HORIZONTALLY WHETHER EACH DEVICE SEEM SUITABLE FOR NAVIGATION AND/OR ANNOTATION TASKS, VERTICALLY WHETHER THE TIED GESTURAL INPUT VS VISUAL OUTPUT MODALITIES RELATION IS DIRECT OR INDIRECT.

1) Visualization: The earlier visualization techniques regarding annotation were often offered by the recording device itself: sensor plots, video films, audio tapes, and so on... The closest task to multimedia content annotation is multimedia edition, notably with audio and video sequencers that can record signals, segment them, apply effects on them and realign them along the timeline.

Lots of techniques dedicated to time series have been proposed so far [3, 33]. Less standard information visualization techniques considering the user perception [63] might improve the task of multimodal annotation, during monitoring of recording processes and post-recording analysis. For a more in-depth analysis, different types of plots can help reduce the complexity of multidimensional data spaces and allow visual data mining. Animations between visualization techniques switched during the task may arouse cognitive effects and improve the user's comprehension of the underlying information present within the displayed data [22, 5]. We follow this overview with specificities to some media types we chose to investigate: audio and video.

a) Audio: waveforms...: A survey of waveforms visualization techniques is proposed in [17], using visual variables to display more information than envelope or amplitude, rather: segments, frequency and timbral content, etc... Some advice is offered on how to visualize waveforms under small scale constraints, particularly by neglecting the negative part of the waveform or subtracting it to the positive part so as to overlap both, similar to a half-rectified signal. A regressive variation on these "mirrored graphs" called "n-band horizon graphs" [21], effectively reducing the height of time-series while keeping readability of information at high zoom factors, seems particularly useful for multitrack timeline representations.

b) Video: keyframes...: Video content is often represented by its frames or keyframes in various ways:

- all frames aligned in time horizontally;
- a subset of these sequenced in time and overlapped in location (such "animated GIF" image files serving as thumbnails on video hosting portals such as Archive.org);
- a standard video player where all frames are displayed on the same location, overlapped in time.

Other spatio-temporal content-specific techniques have been for video signals, for instance "MotionGrams" [26] or "slit/video scanning" [40], particularly suited to videos featuring movement of recurring elements in the scene (again, for instance, dancers videos, among other examples in interactive arts [40]). Lee et al. proposed several keyframe browsing prototypes [36] characterized along a 3D design space: *layerdness* (single/multiple layer with/without links), *temporal orientation* (relative/absolute/none) and *spatial vs. temporal visualization*.

2) Gestural input: Keyboards and mice interaction is still standard for most desktop applications [39], key bindings appears to be the fastest way of triggering momentary or ranged annotation when navigating on the signals with a constant playback speed [18]. Pen have been used by human people to annotate graphics and plots long before their recent computerized versions, now free-form [30] with styluses [2]. Jog wheels for navigating in audio and video signals have been widely used by experts of audio edition and video montage before multimodal annotation. Multitouch interfaces allow the combination of both navigation and annotation modes using one single gestural input modality. The direct or indirect gestural vs visual relation of the user interface can affect the spatial accuracy and speed of annotation tasks [52]. We have illustrated these concepts in Fig. 3 by representing gestural input modalities illustrated with common associated low-cost controllers.

3) Other possible modalities: As raised in Section II-A, similar sensors can be used to record both the multimedia signals being annotated with the annotation tool and the multimodal signals used in the user interface from the tool, thus such modalities used for multimodal emotion recognition [61], for instance eye gaze could be used to improve the location of annotations and predict regions of interest for the user so as to better layout notifications; while voice input with speech recognition could help produce instant user-defined tags or accurate dubbing of meeting recordings.

C. Rapid Prototyping

1) Scripted/textual versus visual programming: Signal processing and engineering specialists often use scripted/textual programming for their prototypes (for instance using Matlab) and they optionally switch to visual programming dataflow environments when realtime prototyping is of concern (with LabVIEW, Simulink, etc...). We believe that blending both approaches is convenient for the process of designing and prototyping the multimodal user interface of our adapted tool: visual programming gives a visual representation by itself of the underlying interaction pipeline, quite practical for exchanging design cues, while textual programming is quicker a designing simple and fast procedural loops, among other advantages.

2) Visual Programming Environments for Multimodal Interfaces:

a) Existing visual programming tools: The number of multimodal prototyping tools and frameworks, dedicated to gestural input or generic towards most multimodal interfaces, has been increasing over the last two decades, yet none of them has been accepted so far as an industry standard. Among the vast availability, we would like to cite some that are still accessible, alphabetically: HephaisTK [10], ICon [9] and the post-WIMP graphical toolkit MaggLite [24] based on top of it, OpenInterface [38] (with its OIDE [54] and Skemmi [35] visual programming editors), Squidy Lib [31].

Data flow environments such as EyesWeb [25], PureData [45] and Max/MSP [7] benefit from their anteriority in comparison with these multimodal prototyping tools, as they often provide more usable visual programming development environments. Some of the authors of this report have been successfully using PureData as a platform for rapid prototyping of gestural interfaces [14]. A notable nice feature from these environments that could be repurposed in the ones targeted for multimodal user interfaces: the "multi-fidelity" patch/pipeline representation modes of Cycling Max/MSP:

1) in "edit" or "patch" mode, the dataflow representation of the pipeline, widgets of processing blocks are editable and interconnections apparent between these;
2) in "running" or "normal" mode, widgets from the pipeline are interactive, but interconnections are hidden;
3) in "presentation" mode, widgets are "ideally" positioned as it would be expected from a control GUI and connections are hidden as well.

OpenInterface/Skemmi addresses this issue with designer/developper modes and a non-linear zoom slider while Squidy Lib offers a zoomable user interface.

b) Chosen plaform: OpenInterface (OI): The OpenInterface platform [34] developed by one of the participants, Lionel Lawson, facilitates the rapid prototyping of multimodal interfaces in a visual programming environment. It also eases technically the communication between components written in different development languages (currently: C++, Java, Python, .NET) in Windows and Linux OSes. It already features several input device components (WiiMote, webcams for computer vision, 3D mice) and some gesture recognition components, but misses a few important ones (multitouch screen/tablets, pen tablet) for the scope of our project. We decided to maintain using this platform and implement the missing components.

3) Environments for "GUI" and visualization:

a) Existing tools: Regarding visualization, mostly libraries are available rather than rapid prototyping tools, particularly Prefuse [20] for information visualization or VTK and Visualization Library for 3D computer aided design or medical visualization. The Processing Development Environment (PDE) [15] simplifies the development in Java and goes further than visualization by providing other libraries for gestural input for instance. Emerging libraries such as MT4j [13] in Java and PyMT [46] in Python offer high-level "multimedia" widgets with multitouch support, yet customization of widgets still requires some effort. The more recent VisTrails / VisMashup [53] allows visual programming of workflows for data exploration and visualization.

b) Chosen plaform: the Processing Development Environment (PDE): We chose the Processing Development Environment (PDE) [15] since it was already mastered by the participants of the team working on designing new proposals for the graphical user interface of the annotation tool. Additionally, more scalability is offered by this solution for the prototyping: since PDE is written in Java, it is compatible with our chosen rapid prototyping platform, OpenInterface (see Section IV-C.2.b), using proclipsing [44], a bridge to the Eclipse IDE used on top of which the OpenInterface Skemmi editor is built; but it can also be re-integrated into a more standalone MT4j application if only a multitouch interface is chosen for gestural input, hence removing the dependency to OpenInterface.

V. Results

A. A tentative design towards an improved User Interface

1) Design considerations: While testing annotation tools (see Section III), we noticed that the user experience with most of the tools was hindered due to the lack of seamless navigation techniques in lengthy signals, for instance changing the playback speed was awkward, both in terms of user input and visualization; and the related audio feedback was improperly rendered. The first task inherent to annotation is navigation into the multimedia content.

2) Mockups: We believe that a single user interface could be used for both the recording of multimodal signals and the navigation into the recorded multimedia content. Figure 4 illustrates a design proposal that would allow this combination: a standard multi-track view of audio, video, and sensor signal tracks stacked vertically is augmented with a sliding vertical zone, extending the proposal of [17] and [59], where are visualized the current frame being played in video tracks (thus behaving like a video player), and a fisheye view of the audio waveform and sensor signals for audio and sensor tracks; the width of the zone corresponding to the same time frame for all tracks.

When recording, the zone could be located on the right, the remaining space left for visualizing past events. When navigating at a given playback speed, the zone could be located in the middle, leaving evenly proportionate space for future and past events, and restricting head movements from the user, gazing towards the center of the screen (as opposed to following visually the play head from left to right cyclicly in standard multitrack editors), the peripheral view optionally stimulated with highlighted past / future events. For a quick overview of the whole recording, the user could want to slide the zone from left to right or to a desired position as a magnifying tool.

3) Prototype: A fast prototype of the proposed design was developed using the Processing Development Environment (PDE) [15], as illustrated in Figure 5.

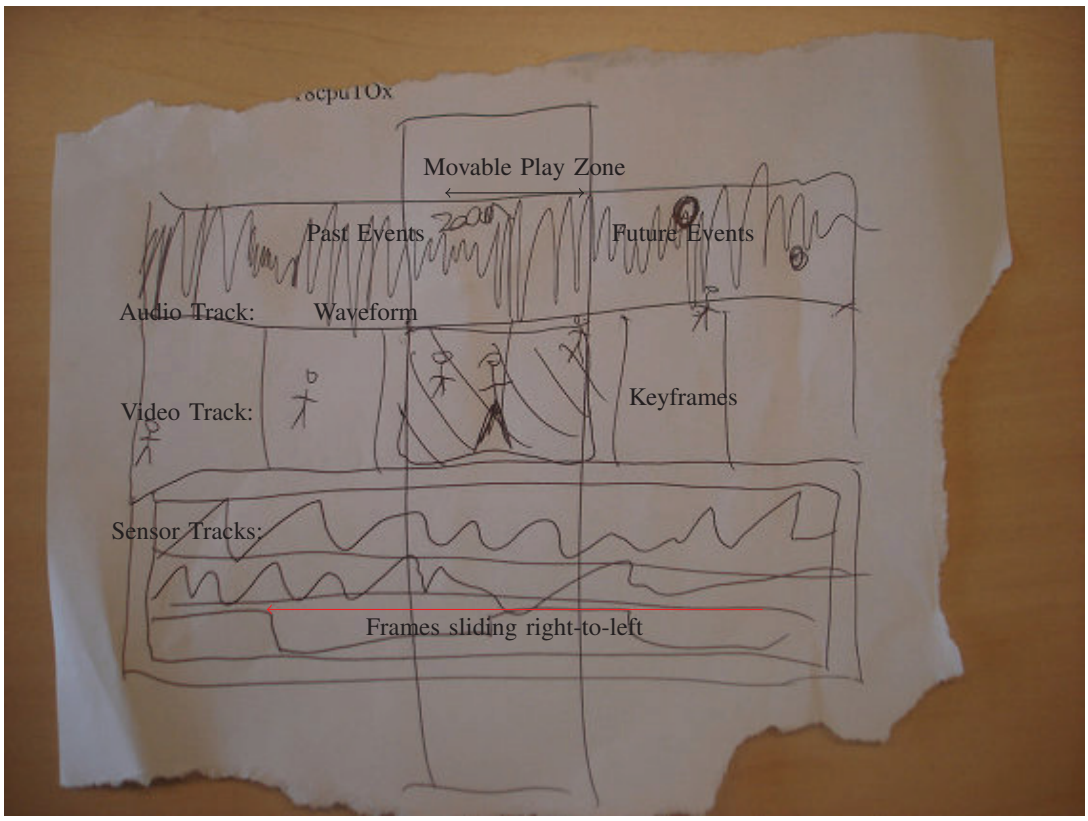


Fig. 4
ANNOTATED PAPER MOCKUP OF OUR PROPOSED USER INTERFACE DESIGN.

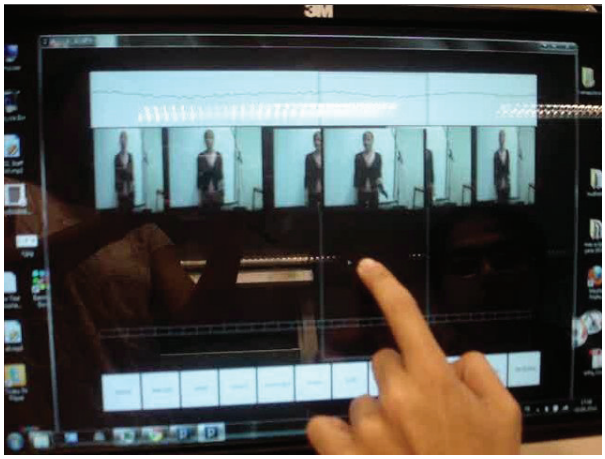


Fig. 5
SCREENSHOT OF THE IMPROVED USER INTERFACE DESIGN PROPOSAL, PROTOTYPED INTO THE PROCESSING DEVELOPMENT ENVIRONMENT.

B. Components for rapid prototyping with OpenInterface (OI)

1) Gestural input: Some device support components were previously available in the OpenInterface platform: the Wii Remote and 3D mice.

For the integration of multitouch devices, 2 options were available:

- capturing WM_TOUCH high-level events from Windows 7 using frameworks such as MT4j [13], but it requires creating applications with the chosen framework;
- accessing low-level events for devices using the Human Interface Device (HID) protocol (cross-platform in theory), reusing code from the GenericHID application for Windows and Linux.

We chose the second option since it also allowed with the same code base to integrate jog wheels (also using the HID protocol).

2) Annotation tool core/engine bindings: In this workshop we decided to adapt the already existing SSI media annotation tool to become a media annotation toolkit with a multimodal user interface. This tool consists of two parts: First, the SSI Media UI, which is a WIMP-GUI based tool to add annotations to audio and/or video data. One can operate it with mouse-clicks and a few keyboard commands. Second, the SSI core component that is responsible for the lower level signal processing. It is used by the SSI Media UI. In the course of the workshop we only adapted the SSI Media UI.

In principle, the given SSI Media toolkit was a prototypical implementation of a media annotation tool. It did not come with a special API that can be utilized from external programs. Therefore, we bundled concrete functionality of the SSI UI Media toolkit into a new interface component. The process of media annotation can be split into three subsequent process steps:

1) create and select annotation tracks (for several annotation channels)
2) segment and reorder segments in one annotation track (includes selection of segments)
3) edit (annotate) segment meta data

This process steps can be performed by the extracted functionality that includes among others start/stop playing of annotation segments, edit of annotations, selection of next/ previous segments, etc. We needed now a way to plug other input modalities to the tool that use this extracted functionality.

Fig. 6
SCREENSHOT OF THE IMPROVED GUI OF THE SSI ANNOTATION TOOL, INTEGRATED INTO THE OPENINTERFACE PLATFORM AS COMPONENT.

We created an OpenInterface component for the adapted SSI Media UI. This OI component allows to use the provided SSI Media functionality by other OI input components. Figure 6 showcases several improvements we applied to its GUI.

C. Testing multimodal pipelines with OpenInterface (OI)

First, we connected the following interaction devices in a new OI project to the SSI OI component. This included a WWI-mote, a 3Dconnexion SpaceNavigator mouse and a Contour Design Xpress jogwheel. Moreover, we created a new speech input OI component for the Julius toolkit [julius.sourceforge.jp/en]. Additionally, we integrated mouse behavior not only by clicking but also with mouse gestures. Each modality (a modality is an interaction device with a dedicated interaction language) was than coupled with specific functionality of the SSI Media toolkit. Not all modalities fit well for the all of the functionality, but this relies to higher-level interaction design. For example, it might be a good idea to select annotation tracks via speech input (command: "next track"). Within such a track one uses mouse gestures to select next and previous segments. When a distinct segment is selected, one uses speech input again to start and stop playing the media (e.g., command: "play segment". And the practicability of the proposed modalities varies. A wii-mote that fits well for arm-based gestures is not the first choice for smaller gestures for media annotation. On the other hand, a jogwheel was invented to improve video editing, thus fitting better for our work. Future work will include research on an improved interaction design, utilizing the "right" modalities for media annotation with the SSI_UI.

VI. FUTURE WORK

A. Integration into MediaCycle, a framework for multimedia content navigation by similarity

MediaCycle is a framework for multimedia content browsing by similarity, developed within the numediart Research Program in Digital Art Technologies, providing componentized algorithms for feature extraction, classification and visualization. The supported media types are: audio (from loops to laughter) [11], video (particularly featuring dancers) [56], images...

This framework already solves some of the issues raised in Section V-A.1 by providing flexible audiovisual engines for the navigation in multimedia content (audio feedback with variable playback speed and visual feedback with cost-effective zoom and animated transitions). Moreover, the interoperation of an annotation timeline (displaying a few elements of the recorded database) with a browser view (displaying the whole database at different levels of segmentation) such as the one already provided by MediaCycle could help compare annotations between recordings and segments. Finally, the use of this framework could help reduce the number of video keyframes by content-based grouping of frames, with a possible scalability against the user-defined zoom factor.

B. Usability testing

We received some feedback from several eNTERFACE participants who had already had to use an annotation tool, regarding their satisfaction with the tool they used. After the setup of a detailed protocol, usability tests based on simple tasks will be performed with the prototype, trying to determine if the user interface improves the annotation efficiency and pleasurability.

VII. Conclusions

We reached a first step towards more usable annotation tools for multimedia content: we raised the problems with current tools and proposed a new design to overcome these issues. The prototype needs to be polished and tested with users to validate the design.

Meaning to produce deliverables available to most people (low-cost, open-source, and so on...) the eNTERFACE way, we developed:

- a free and opensource toolbox, mostly based on cross-platform tools and libraries;
- compatibility with low-cost input devices;
- a starting point to undertake usability testing that demonstrate the validity of the proposed solution.

VIII. Acknowledgements

Christian Frisson works for the numediart long-term research program centered on Digital Media Arts, funded by Région Wallonne, Belgium (grant N°716631).

Ceren Kayalar's PhD Research Project is partly funded by TUBITAK Career Research Grant 105E087 of her advisor, Dr. Selim Balcısoy.

Florian Lingenfelser and Johannes Wagner are funded by the EU in the CALLAS Integrated Project (IST-34800).

We would like to thank the eNTERFACE participants who provided us some feedback on their use of annotation tools and remarks on our design, notably Ismail Arı, Dennis Reidsma [47] and Albert Ali Salah.

We would like to thank all members of the eNTERFACE'10 organizing committee for ensuring a tight workflow (and social events) throughout the workshop.

IX. References

A. Scientific references (books, journals, conferences, workshops)

[2] Maneesh Agrawala and Michael Shilman. "DIZI: A Digital Ink Zooming Interface for Document Annotation". In: *Proceedings of INTERACT*. 2005. URL: http://graphics.stanford.edu/papers/dizi/DIZI.3.pdf. P.: 4.

[3] Wolfgang Aigner et al. "Visual Methods for Analyzing Time-Oriented Data". In: *IEEE Transactions on Visualization and Computer Graphics* 14.1 (2008). Pp. 47–60. URL: http://www.informatik.uni-rostock.de/~ct/Publications/tvcg08.pdf. P.: 4.

[5] Anastasia Bezerianos, Pierre Dragicevic, and Ravin Balakrishnan. "Mnemonic Rendering: An Image-Based Approach for Exposing Hidden Changes in Dynamic Displays". In: *Proceedings of UIST 2006 - ACM Symposium on User Interface Software and Technology*. 2006. Pp. 159–168. URL: http://www.dgp.toronto.edu/~anab/mnemonic/. P.: 4.

[6] Tony Bigbee, Dan Loehr, and Lisa Harper. *Emerging Requirements for Multi-Modal Annotation and Analysis Tools*. Tech. rep. The MITRE Corporation, 2001. URL: http://www.mitre.org/work/tech_papers/tech_papers_01/bigbee_emerging/bigbee_emerging.pdf. P.: 2.

[8] Stefanie Dipper, Michael Götze, and Manfred Stede. "Simple Annotation Tools for Complex Annotation Tasks: an Evaluation". In: *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*. 2004. Pp. 54–62. URL: http://www.ling.uni-potsdam.de/%7Edipper/papers/xbrac04-sfb.pdf. P.: 2.

[11] Stéphane Dupont et al. "Browsing Sound and Music Libraries by Similarity". In: *128th Audio Engineering Society (AES) Convention*. 2010. P.: 7.

[12] L. Dybkjaer and N. O. Bernsen. "Towards general-purpose annotation tools: how far are we today?" In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation LREC'2004*. 2004. URL: http://www.nis.sdu.dk/~nob/publications/LREC2004-annotation-DybkjaerBernsen.pdf. Pp.: 2, 3.

[14] Christian Frisson et al. "DeviceCycle: rapid and reusable prototyping of gestural interfaces, applied to audio browsing by similarity". In: *Proceedings of the New Interfaces for Musical Expression++ (NIME++)*. 2010. ISBN: 978-0-646-53482-4. URL: http://www.educ.dab.uts.edu.au/nime/PROCEEDINGS/papers/Demo%20Q1-Q15/P473_Frisson.pdf. P.: 5.

[16] Patrick Gebhard et al. "Authoring Scenes for Adaptive, Interactive Performances". In: *Proceedings of the ACM AAMAS*. 2003. P.: 2.

[17] Kristian Gohlke et al. "Track Displays in DAW Software: Beyond Waveform Views". In: *Audio Engineering Society Convention 128*. 2010. Pp.: 4, 5.

[18] Joey Hagedorn, Joshua Hailpern, and Karrie G. Karahalios. "VCode and VData: Illustrating a new Framework for Supporting the Video Annotation Workflow". In: *Proceedings of AVI*. 2008. Pp.: 2–4.

[19] Björn Hartmann et al. "Authoring Sensor-based Interactions by Demonstration with Direct Manipulation and Pattern Recognition". In: *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI)*. 2007. P.: 2.

[20] Jeffrey Heer, Stuart K. Card, and James A. Landay. "Prefuse: A Toolkit for Interactive Information Visualization". In: *ACM Human Factors in Computing Systems (CHI)*. 2005. URL: http://vis.berkeley.edu/papers/prefuse/. P.: 5.

[21] Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. "Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations". In: *Proceedings of CHI*. 2009. P.: 4.

[22] Jeffrey Heer and George Robertson. "Animated Transitions in Statistical Data Graphics". In: *IEEE Information Visualization (InfoVis)*. 2007. URL: http://vis.berkeley.edu/papers/animated_transitions/. P.: 4.

[23] A. Heloir, M. Neff, and M. Kipp. "Exploiting Motion Capture for Virtual Human Animation". In: *Proceedings of the Workshop "Multimodal Corpora: Advances in Capturing, Coding and Analyzing Multimodality" at LREC-2010*. 2010. URL: http://embots.dfki.de/doc/Heloiretal10.pdf. P.: 2.

[26] Alexander Refsum Jensenius. "Using Motiongrams in the Study of Musical Gestures". In: *ICMC 2006*. 2006. URL: http://www.hf.uio.no/imv/forskning/forskningsprosjekter / musicalgestures / publications/pdf/jensenius-icmc2006.pdf. P.: 4.

[27] Ceren Kayalar, Emrah Kavlak, and Selim Balcısoy. "A User Interface Prototype For A Mobile Augmented Reality Tool To Assist Archaeological Fieldwork". In: *SIGGRAPH'08: ACM SIGGRAPH 2008 Posters*. 2008. URL: http://students.sabanciuniv.edu/~ckayalar/siggraph08_poster_kayalar_kavlak_balcisoy.jpg. P.: 2.

[29] Michael Kipp. "Spatiotemporal Coding in ANVIL". In: *Proceedings of the 6th international conference on Language Resources and Evaluation (LREC-08)*. 2008. URL: http:

//embots.dfki.de/doc/Kipp08_Anvil.pdf. Pp.: 2, 3.

[30] Nicholas Kong and Maneesh Agrawala. "Perceptual Interpretation of Ink Annotations on Line Charts". In: *Proceedings of UIST*. 2009. P.: 4.

[32] Johannes Kopf et al. "Capturing and viewing gigapixel images". In: *SIGGRAPH'07: ACM SIGGRAPH 2007 papers*. San Diego, California 2007. P.: 2.

[33] Rony Kubat et al. "TotalRecall: Visualization and Semi-Automatic Annotation of Very Large Audio-Visual Corpora". In: *Ninth International Conference on Multimodal Interfaces (ICMI 2007)*. 2007. URL: http://www.media.mit.edu/cogmac/publications/kubat_icmi2007.pdf. P.: 4.

[34] Jean-Yves Lionel Lawson et al. "An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components". In: *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems (EICS'09)*. 2009. P.: 5.

[36] Hyowon Lee et al. "Implementation and analysis of several keyframe-based browsing interfaces to digital video". In: *in Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*. Springer, 2000. Pp. 206–218. P.: 4.

[37] Michael Lew. "Live Cinema: Designing an Instrument for Cinema Editing as a Live Performance". In: *Proceedings of New Interfaces for Musical Expression (NIME)*. 2004. URL: http://nime.org/2004/NIME04/paper/NIME04_3A03.pdf. P.: 2.

[39] Bill Moggridge. *Designing Interactions*. The MIT Press, 2007. ISBN: 9780262134743. URL: http://www.designinginteractions.com. P.: 4.

[40] Michael Nunes et al. "What Did I Miss? Visualizing the Past through Video Traces". In: *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'07)*. 2007. URL: http://grouplab.cpsc.ucalgary.ca/grouplab/uploads/Publications/Publications/2007-VideoTraces.ECSCW.pdf. P.: 4.

[42] Andrei Popescu-Belis. "Multimodal Signal Processing: Theory and applications for human-computer interactions". In: ed. by Jean-Philippe Thiran, Ferran Marqués, and Hervé Boulard. Elsevier, 2009. Chap. Managing Multimodal Data, Metadata and Annotations: Challenges and Solutions, pp. 207–228. ISBN: 978-0-12-374825-6. P.: 2.

[43] Yannick Prié, Olivier Aubert, and Bertrand Richard. "Démonstration: Advene, un outil pour la lecture active audiovisuelle". In: *IHM'2008*. 2008. URL: http://liris.cnrs.fr/advene/doc/advene-demo-ihm08.pdf. Pp.: 2, 3.

[47] Dennis Reidsma. "Annotations and Subjective Machines — of annotators, embodied agents, users, and other humans". PhD thesis. University of Twente, 2008. DOI: 10.3990/1.9789036527262. P.: 8.

[48] Dennis Reidsma, Dennis H. W. Hofs, and Natav sa Jovanovi'c. "Designing Focused and Efficient Annotation Tools". In: *Measuring Behaviour*. Ed. by L. P. J. J. Noldus et al. Wageningen, NL 2005. Pp. 149–152. ISBN: 90-74821-71-5. URL: http://doc.utwente.nl/65561/1/reidsmaMB05.pdf. P.: 2.

[49] Richard Rinehart. "The Media Art Notation System: Documenting and Preserving Digital/Media Art". In: *Leonardo* 40.2 (Apr. 2007). 2. Pp. 181–187. P.: 2.

[50] Katharina Rohlfing et al. *Comparison of multimodal annotation tools*. Tech. rep. Gesprächsforschung - Online-Zeitschrift zur verbalen Interaktion, 2006. URL: http://www.gespraechsforschung-ozs.de/heft2006/tb-rohlfing.pdf. P.: 2.

[51] Natalie Ruiz, Fang Chen, and Sharon Oviatt. "Multimodal Signal Processing: Theory and applications for human-computer interaction". In: ed. by Jean-Philippe Thiran, Ferran Marqués, and Hervé Boulard. Elsevier, 2009. Chap. Multimodal Input, pp. 231–256. ISBN: 978-0-12-374825-6. P.: 3.

[52] Dan Saffer. *Designing Gestural Interfaces*. O'Reilly Media, Inc., 2009. ISBN: 978-0-596-51839-4. URL: http://www.designinggesturalinterfaces.com/. P.: 4.

[53] Emanuele Santos et al. "VisMashup: Streamlining the Creation of Custom Visualization Applications". In: *IEEE Visualization*. 2009. URL: http://www.cs.utah.edu/~juliana/pub/mashup-vis2009.pdf. P.: 5.

[55] Noah Snavely, Steven M. Seitz, and Richard Szeliski. "Photo tourism: exploring photo collections in 3D". In: *SIGGRAPH'06: ACM SIGGRAPH 2006 Papers*. Boston, Massachusetts 2006. ISBN: 1-59593-364-6. DOI: http://doi.acm.org/10.1145/1179352.1141964. P.: 2.

[56] Damien Tardieu et al. "An interactive installation for browsing a dance video database." In: *IEEE International Conference on Multimedia & Expo*. 2010. Pp.: 2, 7.

[61] Johannes Wagner, Elisabeth André, and Frank Jung. "Smart sensor integration: A framework for multimodal emotion recognition in real-time". In: *Affective Computing and Intelligent Interaction (ACII 2009)*. 2009. URL: http://mm-werkstatt.informatik.uni-augsburg.de/files/publications/261/ssi_acii09_camera.pdf. Pp.: 2, 3, 5, 11.

[62] Johannes Wagner et al. "SSI/ModelUI - A Tool for the Acquisition and Annotation of Human Generated Signals". In: *DAGA*. 2010. URL: http://mm-werkstatt.informatik.uni-augsburg.de/files/publications/295/wagner_daga2010.pdf. Pp.: 3, 4.

[63] Colin Ware. *Visual Thinking: for Design*. Interactive Technologies. Morgan Kaufmann, 2008. ISBN: 978-0123708960. P.: 4.

B. Software (annotation tools, rapid prototyping frameworks...)

[1] "Advene (Annotate Digital Video, Exchange on the NEt)". URL: http://www.advene.org. Pp.: 2, 3.

[4] "AmiGram: AMI Graphical Representation and Annotation Module". URL: http://ami.dfki.de/amigram/. P.: 3.

[7] Cycling'74. "Max/MSP". URL: http://www.cycling74.com. P.: 5.

[9] Pierre Dragicevic, Jean-Daniel Fekete, and Stéphane Huot. "ICon (Input Configurator)". URL: http://inputconf.sourceforge.net. P.: 5.

[10] Bruno Dumas. "HephaisTK". URL: http://sourceforge.net/projects/hephaistk/. P.: 5.

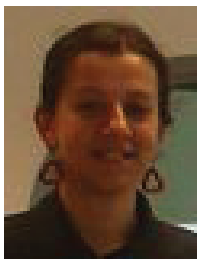
[13] Fraunhofer - Institute for Industrial Engineering. "MT4j - Multitouch for Java™". URL: http://www.mt4j.org. Pp.: 5, 6.

[15] Ben Fry and Casey Reas. "Processing Development Environment (PDE)". URL: http://www.processing.org. P.: 5.

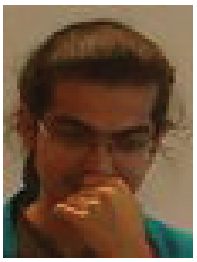
[24] Stéphane Huot and Cédric Dumas. "MaggLite". URL: http://www.emn.fr/x-info/magglite/. P.: 5.

[25] DIST-University of Genova InfoMus Lab. "The EyesWeb XMI (eXtended Multimodal Interaction) platform". Version 5.0.2.0. URL: http://www.eyesweb.org. P.: 5.
[28] Michael Kipp. "ANVIL: The Video Annotation Research Tool". URL: http://www.anvil-software.de. Pp.: 2, 3.
[31] Werner A. König, Roman Rädle, and Harald Reiterer. "Squidy Lib". URL: http://www.squidy-lib.de. P.: 5.
[35] Lionel Lawson and Amro Al-Akkad. "Skemmi, an Eclipse based front-end to OpenInterface Runtime". URL: https://forge.openinterface.org/projects/skemmi/. P.: 5.
[38] Lionel Lawson et al. "The OpenInterface platform". URL: http://www.openinterface.org. P.: 5.
[41] IRI / Centre Pompidou. "Lignes de Temps". URL: http://www.iri.centrepompidou.fr. Pp.: 2, 3.
[44] "proclipsing - Eclipse Processing Development Tools". URL: http://code.google.com/p/proclipsing/. P.: 5.
[45] Miller Puckette and all PureData developers. "PureData". URL: http://www.puredata.info. P.: 5.
[46] "PyMT". URL: http://pymt.txzone.net. P.: 5.
[54] Marcos Serrano and Michael Ortega. "The OI Interaction Development Environment (OIDE)". URL: https://forge.openinterface.org/projects/oide/. P.: 5.
[57] The Technical Group of the Max Planck Institute for Psycholinguistics. "ELAN". URL: http://www.lat-mpi.eu/tools/elan. Pp.: 2, 3.
[58] "VCode & VData: Video Annotation Tools". URL: http://social.cs.uiuc.edu/projects/vcode.html. Pp.: 2, 3.
[59] VeriCorder. "1st Video: Mobile Video Editing Software". URL: http://www.vericorder.com. P.: 5.
[60] Johannes Wagner. "Smart Sensor Integration (SSI)". URL: http://mm-werkstatt.informatik.uni-augsburg.de/ssi.html. Pp.: 2, 3, 11.
[64] David Young. "On The Mark". URL: http://onthemark.sourceforge.net. Pp.: 2, 3.

X. Biographies



Sema Alaçam was born in Malatya, Turkey, in 1981. She studied architecture at Istanbul University. She continued her master degree at Istanbul Technical University (ITU), Institute of Science and Technology, Department of Informatics, Architectural Design Computing Graduate Program between 2005 and 2007. In 2006-2007 fall and spring semesters she has been an erasmus/exchange student at Technical University of Delft, Netherlands, Hyperbody Research Group. Her master thesis, entitled "An interface proposal for collaborative architectural design process", has been supported by BAP-ITU (Scientific Research Projects - Istanbul Technical University) and TUBITAK (Scientific and Technological Research Council of Turkey) Since 2007, she has been working as a full-time research and teaching assistant at Istanbul Technical University, Institute of Science and Technology, Department of Informatics, in the same department, she is a PhD candidate in Architectural Design Computing Graduate Program since 2008.



Emirhan Coşkun was born in Istanbul, Turkey, in 1986. He received his B. Architectural degree from Istanbul Technical University (ITU), Turkey, in 2008, where he studied Architectural Design and Studies. He is currently a M.Sc student at chair for Architectural Design Computing at the same university (Design Computing), working on emergence of the future cities and framework of emergence. He is currently developing a general framework using the Processing Development Environment for the user movements of the people in city and its integration into the city life.

Dominik Ertl received his Bsc. degree of Telematik of the University of Technology Graz in November 2005 and his Dipl.-Ing. degree of Computer Technology of the University of Technology Vienna in November 2007. He joined the Institute of Computer Technology in February 2007 as a project assistant and became a University Assistant in August 2008. Currently, he contributes to the SOFAR project (Semantics and Ontologies for Feedback-driven Adapting Recommender Systems). Previous projects were VIMEM (Virtual Machines for Embedded Multimedia Systems) and CommRob (high-level communication with and among robots). He is writing his dissertation about semi-automatic generation of multimodal user interfaces.

Christian Frisson was born in Thionville, France, in 1982. He received his M. Eng. degree in Acoustics and Metrology from Ecole Nationale Supérieure d'Ingénieurs du Mans (ENSIM) at Université du Maine, France, in 2005. He graduated a M. Sc. inclined towards "Art, Science, Technology (AST)" from Institut National Polytechnique de Grenoble (INPG) and the Association for the Creation and Research on Expression Tools (ACROE), France, in 2006. In October 2006, he joined the Communication and Remote Sensing Lab (TELE) of Université catholique de Louvain (UCLouvain), Belgium, and started his PhD with Prof. Benoît Macq in 2008. From September 2010 on, he joined the Circuit Theory and Signal Processing Lab (TCTS) lab from University of Mons (UMons), to pursue his PhD studies with Professors Thierry Dutoit (UMons/TCTS) and Jean Vanderdonckt (UCLouvain-ISYS). He has been a fulltime contributor to the numediart Reseach Program on Digital Art Technologies since 2008.

His research interests feature: user interface design, tangible gestural input, information visualization, rapid prototyping...

Ceren Kayalar was born in Izmir, Turkey, in 1983. She received her BS degree in Computer Engineering from Dokuz Eyll University, Izmir, in 2004; and her MS degree in Computer Science from Sabancı University, Istanbul, in 2006. She is a PhD candidate in Computer Graphics Laboratory (CGLab), Sabancı University. Currently, she is focused on virtual label placement and visualization of labels according to the user's interests and context in mobile augmented reality.

Her research interests include mobile augmented reality, human computer interaction, context-aware computing.

Jean-Yves Lionel Lawson was born in Cotonou (Benin) in 1982. He holds a Master's degree in Computer Science Engineering and a PhD's degree in Applied Sciences from the Université catholique de Louvain (UCLouvain), Louvain-la-Neuve (Belgium). His research interests are in the fields of Software Engineering, Signal Processing and Human-Computer Interactions.

Florian Lingenfelser graduated as a Master of Science in Informatics and Multimedia from the University of Augsburg, Germany, in 2009. Currently he is a PhD student at chair for Multimedia Concepts and Applications Lab of the University of Augsburg, working on multimodal signal processing, multimodal fusion methods and machine learning techniques. He is also co-developing a general framework for the integration of multiple sensors into multimedia applications called Smart Sensor Integration (SSI) [61, 60].

Johannes Wagner graduated as a Master of Science in Informatics and Multimedia from the University of Augsburg, Germany, in 2007. He is currently PhD student at chair for Multimedia Concepts and Applications Lab of the same University, working on multimodal signal processing in the framework of FP6 IP CALLAS. He is currently developing a general framework for the integration of multiple sensors into multimedia applications called Smart Sensor Integration (SSI) [61, 60].

Looking Around with Your Brain in a Virtual World

Danny Plass-Oude Bos, Matthieu Duvinage, Oytun Oktay, Jaime Delgado Saa, Huseyin Gürüler,
Ayhan Istanbullu, Marijn van Vliet, Bram van de Laar, Mannes Poel,
Linsey Roijendijk, Luca Tonin, Ali Bahramisharif, Boris Reuderink

Abstract— **Offline analysis pipelines have been developed and evaluated for the detection of covert attention from electroencephalography recordings, and the detection of overt attention in terms of eye movement based on electrooculographic measurements. Some additional analysis were done in order to prepare the pipelines for use in a real-time system. This real-time system and a game application in which these pipelines are to be used were implemented. The game is set in a virtual environment where player is a wildlife photographer on an uninhabited island. Overt attention is used to adjust the angle of the first person camera, when the player is tracking animals. When making a photograph, the animal will flee when it notices it is looked at directly, so covert attention is required to get a good shot. Future work will entail user tests with this system to evaluate usability, user experience, and characteristics of the signals related to overt and covert attention when used in such an immersive environment.**

Index Terms— **Multimodal interaction, brain-computer interfacing, covert attention, eye tracking, electroencephalography, electrooculography, virtual environment, usability, user experience.**

I. Introduction

So far, most brain-computer interfaces seek to replace traditional input modalities, like mouse or keyboard. However, current electroencephalography-based brain-computer interfaces (EEG-based BCIs) have considerable problems: low speed, low detection accuracies which varies highly between users, low bandwidth, sensitivity to noise and movement, often requiring training, and expensive and cumbersome hardware [1]. These make it difficult to make such BCIs an interesting input method for able-bodied users.

Allison et al. mention a number of considerations for BCI applications for this healthy user group [1]. In this report we touch upon some of them (extending the term BCI to interface using neurophysiological signals):

- **Hybrid BCI**: using BCI in combination with other input signals, either as independent command signal or as a modifier of commands from other inputs.
- **Induced disability**: in circumstances where conventional interfaces are not usable, BCI could function as a replacement, or when they provide not enough bandwidth, BCI could function as an extra input channel.
- **Mapping between cognition and output**: make systems natural in their use by letting the system respond in a way that corresponds to what the user would expect. The interaction does not only consist of the system response however, but also of the user action [2]. Therefore, we would like to propose to extend this definition to include: to use brain activity or mental tasks that come naturally given the situation. This ensures that the system is most intuitive in the interaction, requiring no user learning or memorization.
- **Accessing otherwise unavailable information**: some processes have no outside expression (whether it is just a mental process, or the user is purposefully trying to inhibit such expressions), but could be detected from brain signals.

We developed a system that makes use of naturally-occurring neurophysiological activity to augment the user interaction with a virtual environment, which already uses conventional mouse and keyboard controllers, in a natural way. The main mode of feedback from any computer system is visual, through the computer screen, thus when looking for natural interaction it makes sense to look into tasks that are related to vision: overt and covert attention. Jacob and Karn mention that it is quite difficult to have the system respond to eye gaze in a natural way, which also happens in the real world [2]. The only example they give is human beings: people respond to being looked at, or what other people are looking at. In our prototype, we use this natural response by letting an animal flee when looked at directly. This induces a situational disability (animals cannot be looked at directly), which is solved by using covert attention to get a good view of the creature. But we also show another option for the natural mapping of eye input: when we move our eyes, our view changes. This natural mapping can be translated to adjusting a first person camera in a virtual environment based on the user's eye movement.

Our report will first dive into covert and overt attention, providing background information, the design and evaluation of the pipelines for signal processing and classification, and answering issues related to the use of these pipelines in an online, real-time setting. After this, the whole system is described, with the game application in particular, followed by a description of the online user evaluation experiments we plan to do.

II. Covert Attention

Covert attention is the act of mentally focusing on a target without head or eye movements [3]. While overt attention is said to be an indication of place of focus, covert attention is a possible confound. By detecting both, all options for spatial attention are covered. There is also a theory that covert attention guides saccadic movement, and that it is possibly a mechanism to scan the visual field for points of interest [4].

Offline experiments have shown that when attention is directed to the left visual hemifield, alpha activity decreases in the right posterior hemisphere while simultaneously increasing in the left hemisphere (and vice versa) [5]–[8]. It is also shown in [9]–[11] that not only left-right but also other directions of covert attention are strongly correlated with the posterior alpha.

Covert attention was measured using EEG. EEG and fNIRS are the most suitable methods for healthy users at the moment, because no

D. Plass-Oude Bos, W.M. van Vliet, B.L.A. van de Laar, M. Poel, and B. Reuderink are with HMI group, EEMCS Faculty, University of Twente, The Netherlands. M. Duvinage is with TCTS Lab, Electrical Engineering University of Mons, Belgium. M.O. Oktay is with Electronics and Communications, Corlu Engineering, Namik Kemal University, Turkey, and was partially supported by TUBITAK Research Grant No. 109E202. J. Delgado Saa is with Faculty of Engineering and Natural Sciences, VPA Lab, Sabanci University, Turkey, and IEE Group of Robotics and Intelligent Systems, Universidad del Norte, Colombia. H. Gürüler is with Department of Electronics and Computer Education, Faculty of Technical Education, Mugla University, Turkey. A. Istanbullu is with Faculty of Computer Engineering, Balikesir University, Turkey. L. Roijendijk is with Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, Nijmegen, The Netherlands. L. Tonin is with CNBI, École Polytechnique Fédérale de Lausanne, Switzerland. Ali Bahramisharif is with Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, and Radboud University Nijmegen, Institute for Computing and Information Sciences, both in Nijmegen, The Netherlands.

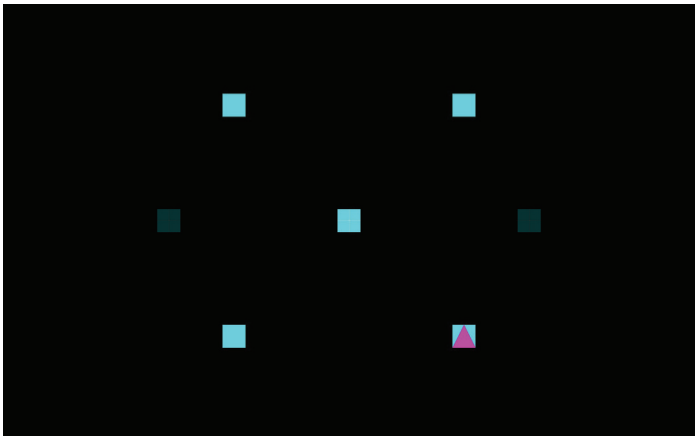


Fig. 1
COVERT ATTENTION SCREEN WITH FIXATION POINT IN THE CENTER, POTENTIAL TARGET SQUARES (DISTRACTORS), AND THE ARROW ON THE ACTUAL TARGET. THE DARKER SQUARES TO THE LEFT AND RIGHT WOULD NORMALLY NOT BE VISIBLE, BUT INDICATE THE ALTERNATIVE FIXATION POSITIONS.

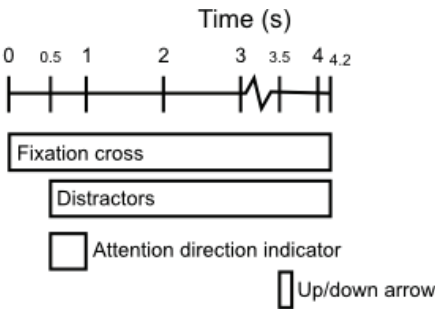


Fig. 2
DURING A TRIAL FIRST THE FIXATION CROSS IS SHOWN, THEN THE DIAGONAL POSITIONS APPEAR, AFTER WHICH THE FOCUS POSITION FOR COVERT ATTENTION IS INDICATED. AFTER A LITTLE WHILE AN UP OR DOWN ARROW IS SHOWN IN THE FOCUS POSITION. THE PARTICIPANT THEN PRESSES THE CORRESPONDING BUTTON.

surgery is required, the equipment can be used outside of a laboratory setup, and the equipment is relatively portable and affordable [1].

This section evaluates a number of potential pipelines, but also another important question: whether this correlation with posterior alpha depends on if a subject fixates centrally or if the same pattern will be observed irrespective of the location the subject's fixation point. While a central fixation point has been the norm in clinical laboratory experiments, in a practical application, this may only rarely be the case. Finally, some other research questions that are relevant for the online situation were looked into: what directions can we detect, how many trials are needed for training, and how long the trial window needs to be for classification?

A. Methods

The experiment is covert attention to the four directions of visual hemifields with three different fixation points. The task is to fixate at each fixation point in the screen which is 70 cm away from the eye of the subject and covertly attend to the direction of the pre-specified arrow. See Figure 1 for a screen shot of the situation. There are three fixation points: left, middle, and right, with six degrees of visual angle distance between them. The target focus can be one of 5 positions: either the fixation point itself (neutral), or one of the four diagonal directions. The focus targets were placed diagonally as earlier research indicated that this is best discriminable [11].

Fifty trials were recorded for each of these conditions consisting of a fixation position and target position. A trial starts with half a second showing the fixation cross, then for half a second the focus position for covert attention is indicated with a yellow circle inside one of the five potential positions. The other positions remain visible as distractors. After a period of 2 seconds plus a random duration of up to half a second, an up or down arrow is shown in the focus position. The participant then has a short period of time to press the corresponding arrow button (arrow up or down). This task ensures that the focus area is relevant to the participant, which may increase the effect on the brain activity for this paradigm. The trials were split up in five blocks, each containing ten repetitions for each condition in randomized order. The breaks in between blocks lasted until the participant pressed a key to continue.

Brain activity is measured during the task using the BioSemi ActiveTwo EEG system, at 512 Hz sampling frequency, with 32 electrodes according to the montage shown in Figure 3. Electrooculogram (EOG) was also recorded to control for confounds in eye movements.

In total datasets were recorded for 8 participants, but for analysis the first two were left out because of marker issues. The last two sets were recorded at a late stage in the project, and thus were not used for every analysis that was conducted.

B. Results

a) Which pipeline performs best?: The four pipelines that were tested were:

Pipeline CA1:

- channels: occipito-parietal
- window: 0.5-2.0sec relative to focus indication stimulus
- feature extraction: CAR, bandpower 9-11Hz STFT, z-score normalization
- classifier: SVM (error cost: 0.1)

Pipeline CA2 (CA1 with whitening, and different SVM error cost parameter):

- channels: occipito-parietal
- window: 0.5-2.0sec relative to focus indication stimulus
- feature extraction: CAR, whitening, bandpower 9-11Hz STFT, z-score normalization
- classifier: SVM (error cost: 2.0)

Pipeline CA3:

- channels: occipito-parietal
- downsample to 256Hz
- window: 0.5-2.0sec relative to focus indication stimulus
- feature extraction: CAR, bandpass 8-14Hz, whitening, covariance
- classifier: logistic regression

Pipeline CA4 (CA1 with different SVM error cost parameter:

- channels: occipito-parietal
- window: 0.5-2.0sec relative to focus indication stimulus
- feature extraction: CAR, bandpower 9-11Hz STFT, z-score normalization
- classifier: SVM (error cost: 2.0)

Table I shows the performance accuracies per pipeline on average but also per subject. CA3 outperforms all others with 67% and 40% on average on the same datasets for two and four-class classification respectively. As pipeline CA3 was implemented in Matlab and not in Python it cannot be applied in the online situation. So for the game,

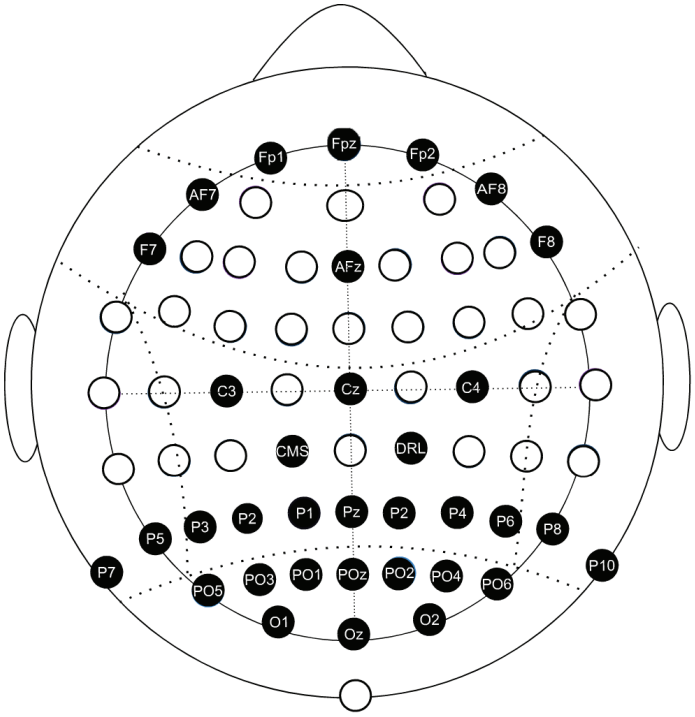


Fig. 3
ELECTRODE POSITIONING FOR EEG MEASUREMENT: 32 ELECTRODES POSITIONED MAINLY ON THE PARIETO-OCCIPITAL AREA AS THIS IS WHERE THE RELEVANT ALPHA MODULATIONS FOR SPATIAL COVERT ATTENTION ARE EXPECTED, AND SOME OTHERS TO LOOK AT ARTIFACTS AND TO OFFER THE POSSIBILITY TO APPLY CERTAIN SPATIAL FILTERS.

we will opt for the second-best pipeline in the two-class case, which is CA4.

b) Does the position of the fixation point matter, with respect to the correlation of focus direction with parietal alpha, and with respect to detection accuracy?: To answer this question, scalp plots were computed for each participant for each fixation position (left, middle, right), showing the relative difference in the alpha band (8–12 Hz) of each diagonal focus direction with the fixation point, see Figures 4-6. A time window from 0.5 to 2 seconds after the cue was used. The scalp plots were averaged over four subjects. The lateralization pattern is in line with what has been shown in literature [8], [9], [11]. As the eyes fixate on a different position, the excitation of the retina remains the same, and the mapping of the image to the occipital cortex is not expected to change. However, surprisingly, the patterns are a bit different for the different blocks, showing a migration of the alpha sources from one side to the other.

On average, there did not seem much of an accuracy difference between each of the fixation point conditions (28%, 30%, 32% and 30% for left, center, right, and pooled fixation points). When looking at our best participant however, we see an increase for the center fixation: 36% for left and right, 40% for pooled, but 45% for center fixation cross only.

c) Which directions can be detected?: Results based on datasets recorded from 4 different participants analyzed with pipeline CA3 indicate a performance above random. For a 4-class situation (each of the four directions) yields a 40% performance accuracy on average, and 52% on our best participant. The samples for the three different fixation points were pooled, so the classes indicate the covert attention direction relative to fixation. Random for four classes would have been 25%. For the two-class situation the bottom and top targets were

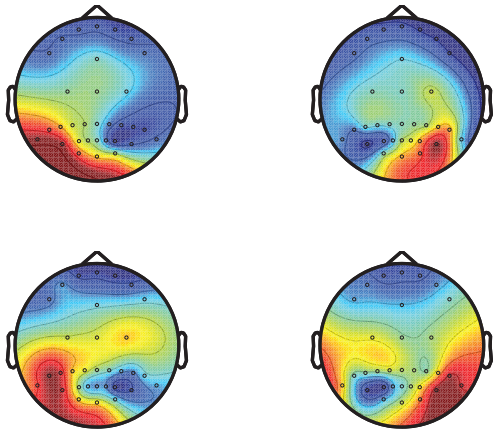


Fig. 4
RELATIVE DIFFERENCES OF EACH FOCUS DIRECTION WITH RESPECT TO THE FIXATION POSITION, WITH THE FIXATION ON THE LEFT.

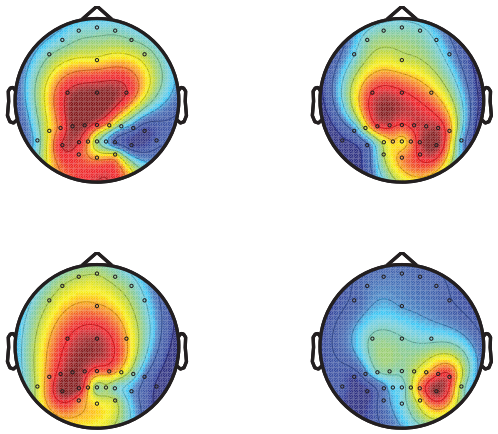


Fig. 5
RELATIVE DIFFERENCES OF EACH FOCUS DIRECTION WITH RESPECT TO THE FIXATION POSITION, WITH THE FIXATION ON THE CENTER.

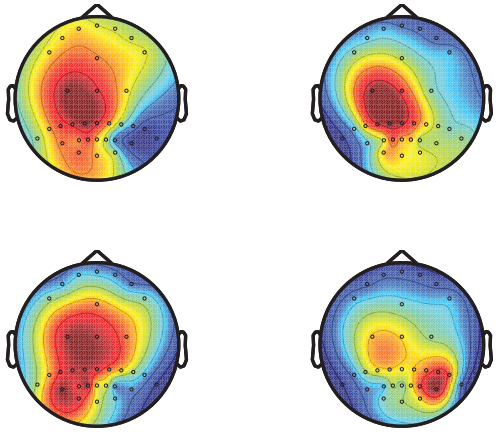


Fig. 6
RELATIVE DIFFERENCES OF EACH FOCUS DIRECTION WITH RESPECT TO THE FIXATION POSITION, WITH THE FIXATION ON THE RIGHT. ALL AVERAGED OVER FOUR SUBJECTS.

TABLE I
PERFORMANCE ACCURACIES OF THE COVERT ATTENTION PIPELINES PER PARTICIPANT. STANDARD DEVIATIONS OF THE PERFORMANCE SCORES ARE BETWEEN PARENTHESES.

4 classes	CA1	CA2	CA3	CA4
S3	32%	32%	33%	31%
S4	35%	30%	31%	31%
S5	44%	35%	52%	42%
S6	37%	34%	44%	35%
Avg	37% (4%)	33% (2%)	40% (2%)	35%
2 classes	CA1	CA2	CA3	CA4
S3	62%	60%	62%	60%
S4	61%	59%	57%	59%
S5	71%	71%	78%	85%
S6	66%	65%	72%	62%
Avg	65% (4%)	64% (5%)	67% (2%)	67% (11%)

merged to result in one class with samples to the left, and one class with samples to the right. For this, the average performance accuracy over 4 subjects with pipeline CA3 was 67%, with 78% for our best subject, against a random performance of 50% for two classes. The other pipelines show a similar pattern in performances, albeit slightly below the scores for CA3.

The classification performances for the different pairs of target directions (like top right vs. top left) were also analyzed. This confirms the information from literature that diagonally opposing targets (top left vs down right, and top right vs down left) are easier to distinguish than the other pairs.

d) How many trials are needed for training?: Figure 7 shows the two-class detection performance with pipeline CA3 for different training set sizes. The performance within the training set was evaluated using 10-fold cross validation. The plot shows no consistent increase in performance. After a peak at 120 trials, performance drops and flattens out.

e) What is the optimal window size?: For the online situation, preferably, the window size is minimal, because that way the data can be processed faster, which in turn could mean that updates can be computed more frequently. On the other hand, the classification accuracy is expected to be higher for longer window sizes (because you simply have more information).

Windows always start at 0.5 seconds after the stimulus, and then continue for the indicated window duration, except for the two-second window which starts at 0.0 seconds.

f) Does a blocked protocol yield a better performance?: In standard covert attention experiments there is only one fixation point, whereas in our experiment, this fixation point was randomized. To test whether this had unwanted side effects, we recorded one dataset which had the fixation points steady within each block, and one in which within a block this fixation point could jump around. The result was a 75% accuracy for both the blocked and not blocked condition of fixation points using pipeline CA3. Based on this one participant, there does not seem to be a difference between the two conditions.

C. Discussion and Conclusions

Pipeline CA3 (CAR, bandpass 8–14 Hz, whitening, covariance, logistic regression) performs best, but could not easily be translated from Matlab to Python. For the online situation we therefore decided to use the second best option: CA4, using CAR, bandpower 9–11 Hz STFT – adjusted to 8–14 Hz, z-score normalization, followed by an SVM classifier.

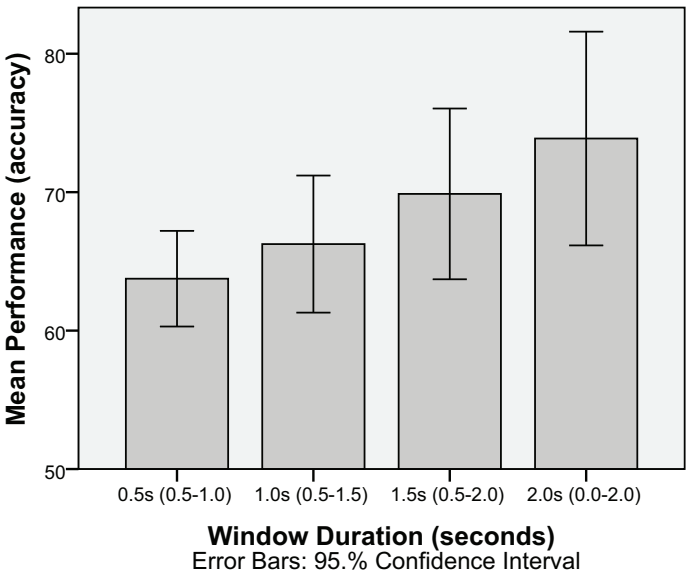


Fig. 8
TWO-CLASS COVERT ATTENTION PERFORMANCE FOR DIFFERENT WINDOW SIZES, AVERAGED OVER 6 SUBJECTS. IT SHOWS AN INCREMENTAL INCREASE FOR LONGER WINDOWS.

Different fixation points (left, middle, right) did not seem to have a significant impact on classification performance. When looking at the relative difference in parietal alpha between the focus direction and central fixation point, similar spatial patterns show which correspond to what is expected from literature. However, there also seems to be a migration of the alpha sources from one side to the other.

Although the four-class performance is above random, for an online game situation performance should be at a usable level. For this reason we decided to use two-class covert attention in the game.

The number of windows in the training dataset, strangely enough, does not seem to have a large impact on the classification performance. The performance does increase from 20 to trials samples, but after that it drops again, stabilizing around the same performance is is shown at around 90 trials. As this is evaluated with 10-fold cross validation, about 80 trials would be enough if all trials are used.

The larger the trial window, the higher the performance. This is to be expected, but less fortunate for the online situation: the longer the window size, the longer it will take to get feedback on that particular window. However, we did not test beyond a size of two seconds, and the test for two seconds could not start at 0.5 seconds as the other windows did. This makes it possible that there are task-related eye movements in those 0.5 seconds that increase the performance.

However, most of these results are based on relatively little data.

III. EYE MOVEMENT

According to Jacob and Karn, using eye movement provides a number of features that make it an interesting input modality. Eye movements are not as intentional as mouse and keyboard input. This means that it can provide information on an intentional but also on a more subconscious level. A side effect is the Midas Touch problem: not every eye gaze has intentional meaning, so the system should somehow discern what to react to, and what not. Eye movement is faster than other input modalities, and already indicates the user's goal before any other action has been taken. Besides, no user training is required, as the relationship between the eye movement and the display is already established [2].

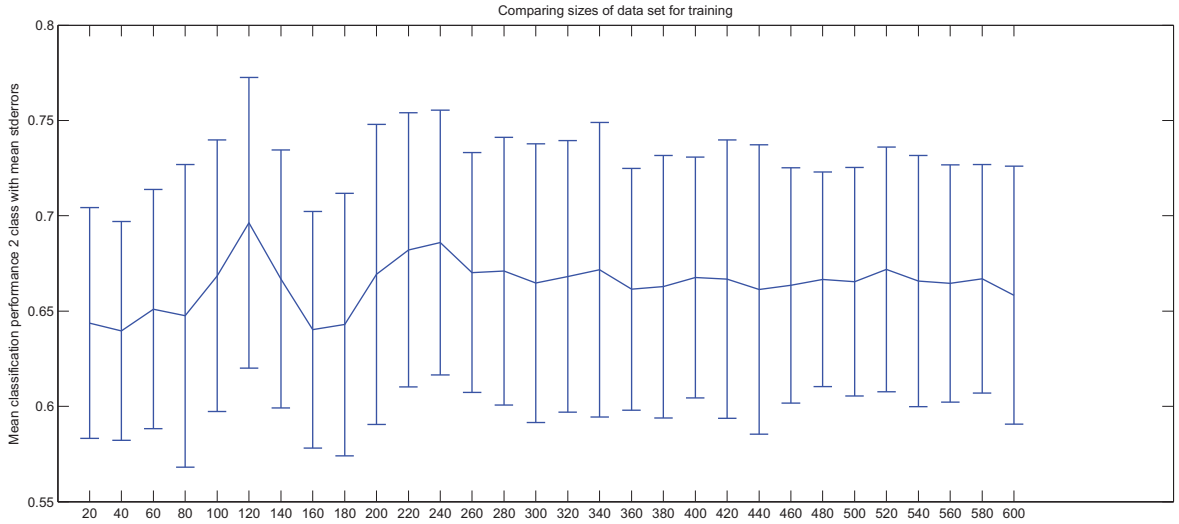


Fig. 7
TWO-CLASS COVERT ATTENTION PERFORMANCE FOR DIFFERENT TRAINING DATASET SIZES, FOR DATASETS 3-6.

Bulling et al. distinguish between the following types of eye movements. Fixations are the stationary states of the eyes during which gaze is focusing on a particular point on the screen, lasting between 100 ms and 200 ms. Saccades are very quick eye movements between two fixations points. The duration of a saccade depends on the angular distance the eyes travel during this movement. For a distance of 20 degrees, the duration is between 10 ms and 100 ms. Eye blinks cause a huge variation in the potential in the vertical electrodes around the eyes, and lasts between 100 ms and 400 ms [12]. For our application, saccades are the most relevant type of movement to detect.

There are a number of methods to determine eye movement or eye gaze, for example with special contact lenses, infrared light reflections measured with video cameras, or with electrodes around the eyes. The last example is also called electrooculography (EOG). The electrodes measure the resting potential that is generated by the positive cornea (front of the eye) and negative retina (back of the eye). When the eye rotates, the dipole rotates as well. By positioning the electrodes around the eyes as shown in Figure 9, one bipolar signal will be an indication of vertical eye rotation and the other for the horizontal axis.

For this system, we decided to use EOG for eye tracking. EOG signal analysis requires very little processing power, and can easily be done in real-time. Although this method is not that suitable for tracking slow eye movements (that occur when following a moving object), for fast saccades it is very robust. EOG can be used in bad lighting conditions (although it works better with good lighting), and in combination with glasses. The participant does not need to be restricted in the orientation to the screen (though for absolute eye gaze, then the position of the head would need to be tracked separately), nor do they have to wear an uncomfortable video camera system firmly mounted on the head [2]. Also, it is easy to incorporate in a wearable and unobtrusive setup [12].

This section explains the pipeline design, an eye blink detection and correction algorithm, the methods for the dataset recording and analysis, details the results of the evaluation, resulting in discussion and conclusions.

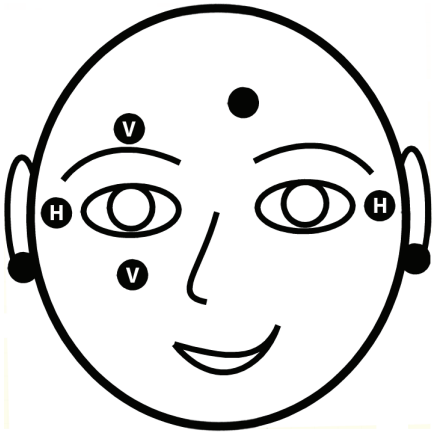


Fig. 9
ELECTRODE POSITIONING FOR EOG MEASUREMENT: BIPOLAR MEASUREMENTS OF TOP MINUS BOTTOM VERTICAL ELECTRODES AROUND THE RIGHT EYE AND RIGHT MINUS LEFT HORIZONTAL ELECTRODES NEAR THE CANTHI.

A. Pipeline

As described in [13], saccade detection can be used to construct an eye-tracker. The pipeline for eye movement is similar for both the vertical and horizontal EOG signals:

1) High pass filter (0.05 Hz) for drift correction which is very strong in the EOG signal.
2) Low pass filter (20 Hz) to reduce high frequency noise without affecting the eye movements.
3) Derivative in order to detect the rapid variations.
4) Thresholding to detect saccades and remove noise.
5) Integration in the saccade range which represents the features.
6) Linear regression between the angle and the integration result.
7) Conversion to x,y position.

The main steps are shown in Figures 10–13 and Figure 14.

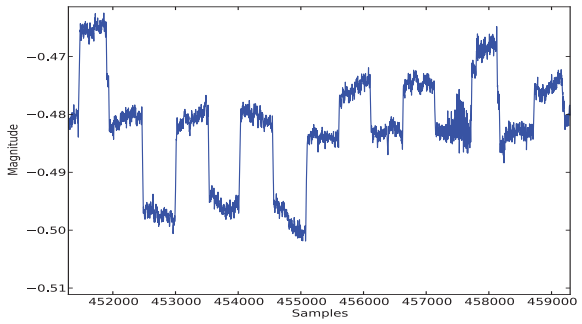


Fig. 10
EOG DATA IS NOISY AND DRIFTS OVER TIME.

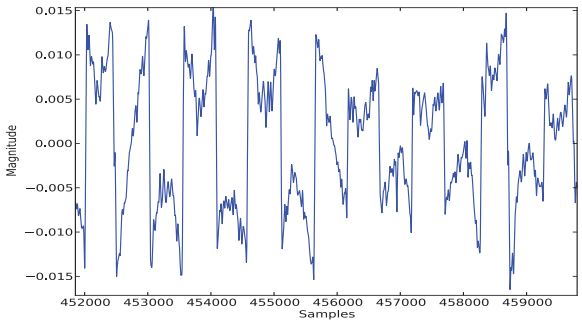


Fig. 11
FILTERED EOG DATA WITHOUT THE DRIFT AND HIGH FREQUENCY NOISE.

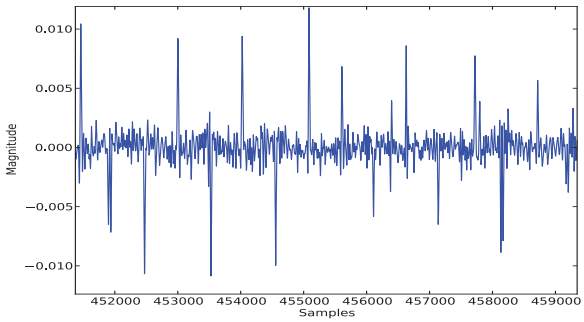


Fig. 12
THE HIGH VALUES OF THE DERIVATIVES INDICATE SACCADES.

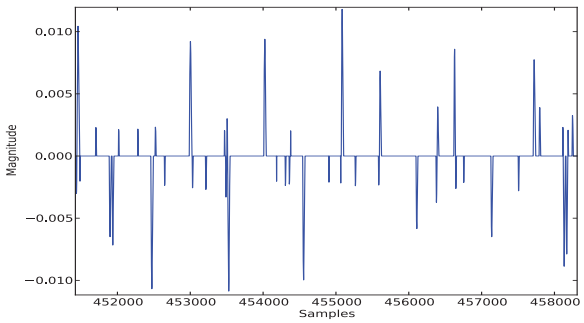


Fig. 13
INTEGRATION OF THE ABOVE-THRESHOLD SACCADE DERIVATIVE PROVIDES THE INPUT FOR THE LINEAR REGRESSION

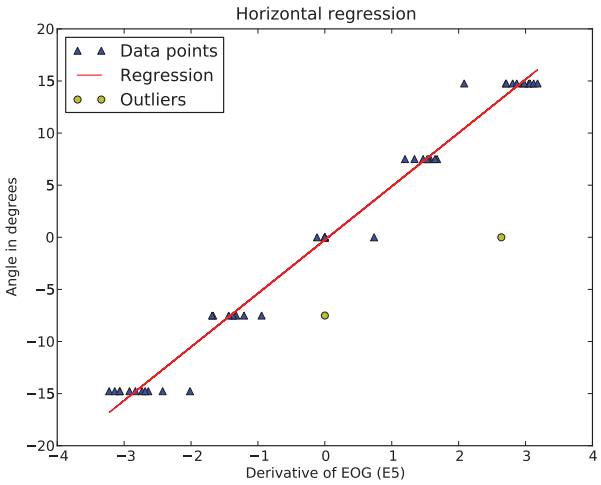


Fig. 14
THE REGRESSION SHOWS A HIGH CORRELATION BETWEEN THE PARAMETER OF EACH SACCADE AND THE JUMP IN ANGLE.

B. Eye Blink Detection

The previous pipeline can be enhanced by eye blink detection and correction. Eye blinks cause in large voltage changes in the vertical EOG signal, which result in a bad estimation of the current eye position 15.

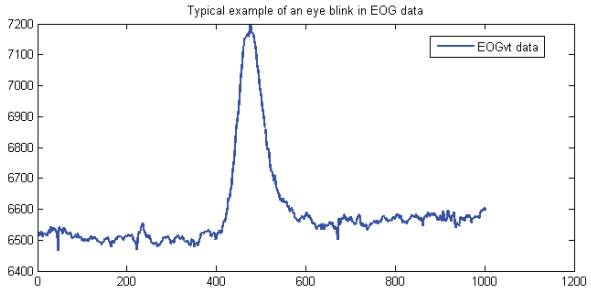


Fig. 15
TYPICAL OCCURRENCE OF AN EYE BLINK IN THE EOG SIGNAL, CHANNEL EOG VERTICAL TOP.

Inspired by [12], a template-based approach to eye blink detection was developed. There were EOG recordings with ten stimulus-based and hence unnatural eye blinks. Based on visual inspection, it was decided to use the EOG channel positioned right above the right eye for eye blink detection. The first five eye blinks were used to construct the initial template of 200 samples long (at a sample frequency of 512 Hz this is about 400 ms). The eye blink examples were aligned by taking a vertical offset such that the mean over time is zero.

This initial template was used to detect more natural eye blinks in the EOG data. Before determining the Euclidean distance between the template and the signal, the template was aligned using a vertical offset which minimizes the distance between the first and last ten samples of the template and the signal fragment under consideration. Figure 16 contains samples of EOG recordings from above the right eye of subject 6d and the corresponding Euclidean distance between the aligned template and the EOG data of the electrode above the right eye. Local minima in the distance below a threshold of 4000 mark the start of an eye blink.

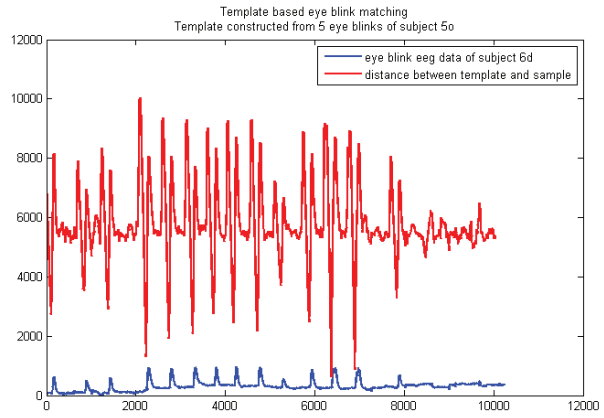


Fig. 16
TEMPLATE MATCHING ON EOG DATA OF SUBJECT 6D. LOCAL MINIMA IN THE DISTANCE BELOW 4000 INDICATE THE START OF AN EYE BLINK .

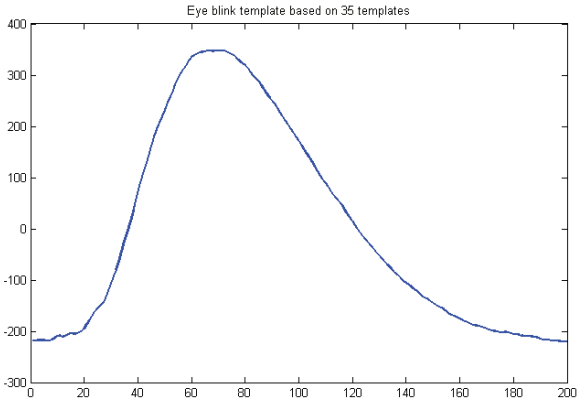


Fig. 17
THE FINAL EYE BLINK TEMPLATE CONSTRUCTED OUT OF 35 EXAMPLES.

These detected eye blinks were used to extend the eye blink examples to 35 and constructing a more realistic final eye blink template afterwards The final eye blink template, see Figure 17, is used to construct an online eye blink detector.

For the online version, we only have a small sliding window to our disposal, which must be larger then the template size. Hence the template matching procedure has to be adapted, in particular the determination of a local minimum in the distance between the aligned template and the signal part. The start of an eye blink is now determined by a switch from decrease to increase (the first derivative changes sign) in the distance function and the constraint that the value of the local minimum is below the threshold. This threshold is dependent on the subject under consideration and can be determined by an online calibration.

Unfortunately, for this preliminary phase in the project, the eye blink detection and correction algorithm was not applied in the eye movement pipeline, because of time constraints.

C. Methods

The offline analysis protocol of the eye movement is twofold. In order to get enough data for training the linear regression, 25 trials were used. Each trial was composed of one target in the center of the screen and one of five possibilities: extreme top, bottom, left, right and center targets. For horizontal and vertical eye movement there are separate pipelines, and the regression is also trained separately – for the pipeline details refer to the Pipeline section above.

For evaluation 100 trials were assessed. Because the system will be used as a kind of eye mouse, the performance evaluation was based on the accuracy of the system at N centimeters maximum deviation from the target. The screen was divided in a 5 by 5 grid, resulting in 25 potential target positions, which were selected randomly. The jump between the center and the target (Figure 19) of each trial is considered correct when the Euclidean distance between the EOG-based estimation point on the screen and the actual point is lower than N centimeters.

These trials were recorded using the BioSemi ActiveTwo hardware, with flat active electrodes positioned according to Figure 9. The distance between the user and the screen was 70 cm.

TABLE II
PERFORMANCE OF THE EYE MOVEMENT PIPELINE PER PARTICIPANT. HORIZONTAL AND VERTICAL ACCURACIES ARE FOR A PRECISION WITHIN 4CM. THE ERROR DISTANCE MEANS AND STANDARD DEVIATIONS ARE MEASURED FROM ACTUAL TARGET POSITION TO REGRESSION RESULT IN HORIZONTAL AND VERTICAL DIRECTIONS.

	Hacc	Herr avg	Herr std	Vacc	Verr avg	Verr std
S4	100.0%	1.0	0.8	94.9%	2.0	6.1
S5	90.9%	2.0	1.6	57.6%	3.9	3.7
S6	70.7%	3.2	3.2	34.3%	7.9	9.4
S8	77.8%	2.4	1.8	51.5%	5.3	4.5

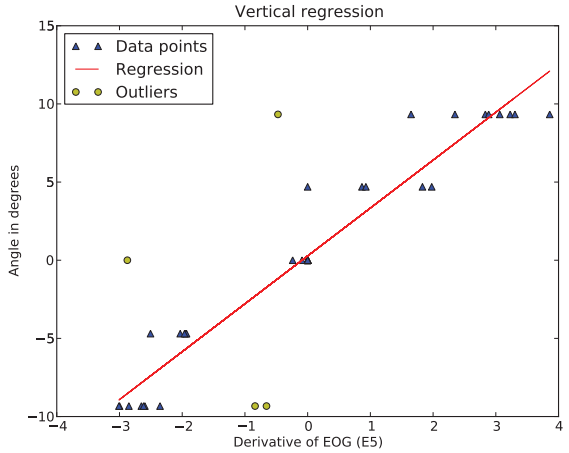


Fig. 18
DATA SHOWS LESS CORRELATION BETWEEN EOG FEATURES AND KNOWN ANGLE CHANGE FOR VERTICAL EYE MOVEMENT.

D. Results

Regarding the horizontal movements, the results are quite good as shown in Table II. Figure 21 shows the precision at N for two participants (S4 and S5). The curve is sharply increasing which shows the precision of this technique. However, regarding the vertical movements, the results are less good (see Table II, Figures 20, 22, and 18, the plots are again based on the data of the two participants S4 and S5).

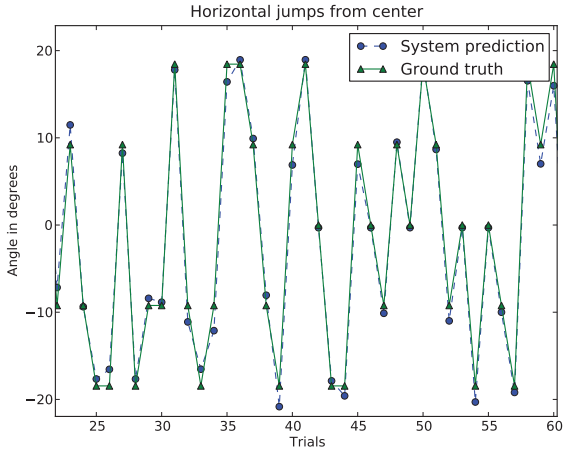


Fig. 19
THE JUMPS BETWEEN THE CENTER AND THE TARGET PROVIDED BY THE SYSTEM AND THE ACTUAL ONES ARE QUITE SIMILAR FOR THE HORIZONTAL AXIS.

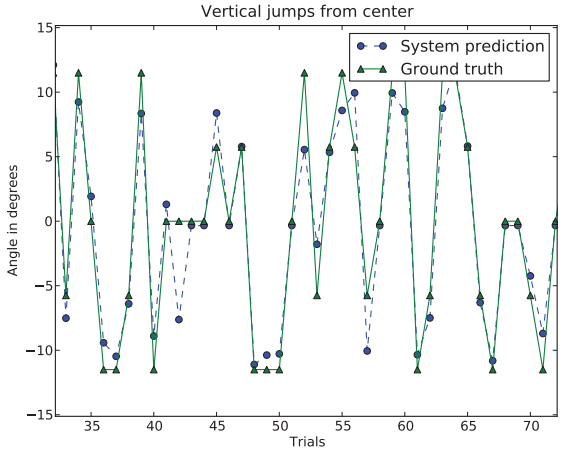


Fig. 20
THE PREDICTIONS OF THE SYSTEM ARE QUITE LESS GOOD THAN FOR THE HORIZONTAL JUMPS.

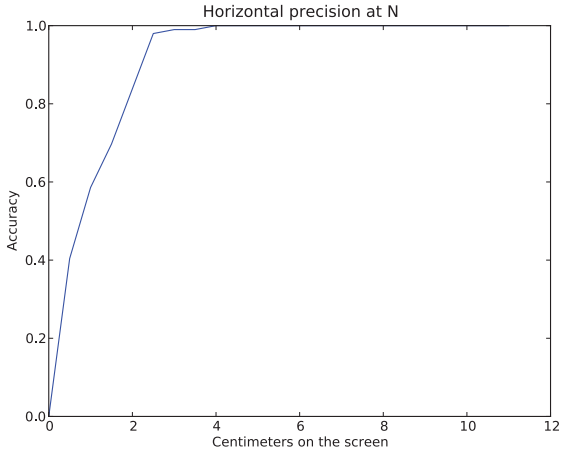


Fig. 21
HORIZONTAL PRECISION AT N CURVE IS SHARP AT THE BEGINNING WHICH IS GOOD.

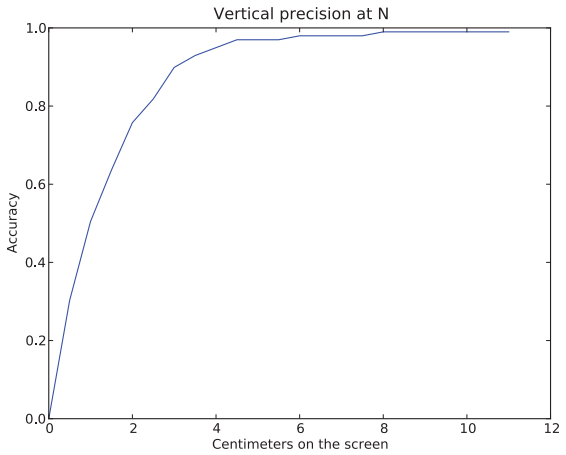


Fig. 22
VERTICAL PRECISION AT N CURVE IS LESS SHARP AT THE BEGINNING THAN THE CORRESPONDING HORIZONTAL CURVE.

E. Discussion and Conclusions

Horizontal eye movement appears to be easily detectable: at a precision within 4 centimeters, the accuracy is about perfect for the best half of the participants. Within 2 centimeters it is about 90%. For vertical eye movement, the performance is a little less good: around 80% for a precision within 2 centimeters.

Visual inspection of the vertical EOG data shows that sometimes there is no sign of the vertical movement when there should be one. Maybe the sensors were not positioned optimally. Also, the vertical distance is smaller than the horizontal distance, meaning that the eyes will turn less degrees, resulting in a smaller potential chance. Moreover, the eye blink detection was not applied in the pipeline. This should also improve performance.

We still need to evaluate the optimal window length for eye movement detection. The window step should be quite short, to give the user the sense of continuous interaction.

IV. APPLICATION AND SYSTEM

The previous sections describe the development and evaluation of the pipelines for covert and overt attention. The goal is to use these covert and overt attention paradigms in a setting which requires real-time input. For this, the pipelines need to require minimum user and system training, the training needs to be integrated within the game, and the signal analysis and classification needs to be fast enough for such an application. As for the application itself: it had to be a game that requires the two paradigms to be used, in a way that is intuitive to the user, and reacts to them in a realistic manner. This way it also serves as proof that such signals can be a valuable addition to a game, and that they allow for new types of games. This section will first describe the game, and then the system bringing it all together.

A. Wild Photoshoot

In the game that was developed, you are a wildlife photographer. On an uninhabited island, you try to make pictures of rare wild animals. But of course wild animals are not that easy to make a good photograph of. First you have to follow animal tracks over the

Fig. 23
A SCREENSHOT OF WILD PHOTOSHOOT IN TRACKING MODE: TWO ANIMAL FOOTPRINTS ARE SHOWING IN THE CENTER-BOTTOM OF THE SCREEN.

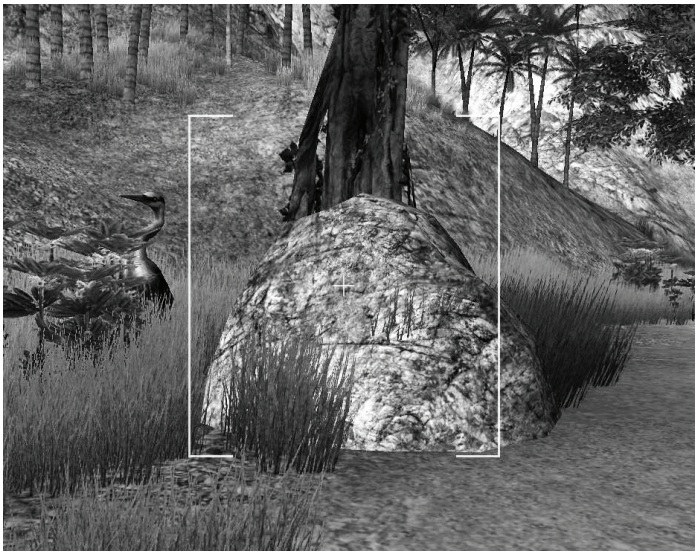


Fig. 24
A SCREENSHOT OF WILD PHOTOSHOOT IN PHOTOSHOOT MODE: THE ANIMAL IS TO THE LEFT, COVERT ATTENTION (FOCUS SQUARE) IS STILL NEUTRAL, AND THE EYE GAZE (FIXATION CROSS) IS ALSO CENTERED.

island to find where the creature is hiding (see Figure 23). Then you go into photoshoot mode in which you try to get a good picture (see Figure 24). When you look directly at the animal, it will flee and you will have to track it again. Thus you have to covertly look at the animal to focus the camera to get your money shot.

Keyboard is used to walk and turn. The system detects eye movement, and adjusts the first person camera to reflect the view angle. When looking towards the left side of the screen, the camera turns towards the left as well, until the user is again looking at the center of the screen. It not only reacts to horizontal, but also to vertical eye movement. The first person camera angle can be adjusted manually by mouse. When in photoshoot mode, covert attention is needed to focus on the animal without looking at it directly. The

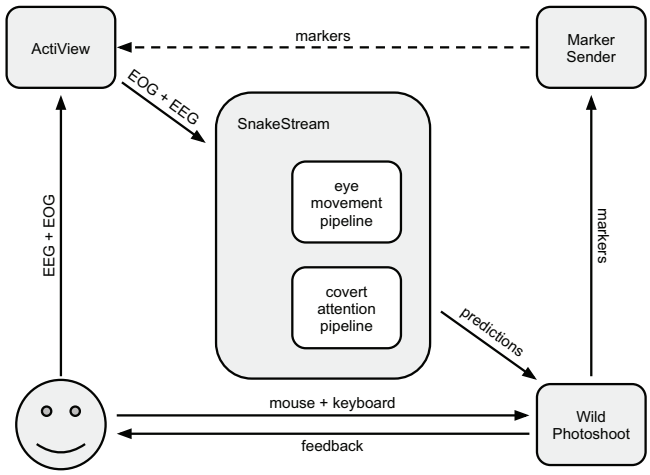


Fig. 25
THE DIFFERENT COMPONENTS AND COMMUNICATION BETWEEN THEM WITHIN THE WILD PHOTOSHOOT SYSTEM.

animal will appear to the left or right of the screen center. The classification of covert attention is divided into two classes: left and right. During the focus period, multiple covert attention classifications are performed. A simple majority vote determines the covert attention direction for the period. If the animal is on the same side as the covert attention, it will result in a nice wildlife photograph. If not, the user will have to try again. After five tries, the animal will notice and flee.

This is a game that uses multiple input modalities: mouse, keyboard, EOG-based overt attention, and EEG-based covert attention. It also creates situational disability for eye movement by letting the animal flee when looked at directly, introducing a natural need for covert attention. The mental tasks for both overt and covert attention come naturally given the situation, and the mapping to system response is based on real-world interaction as well. Finally through covert attention, we access information about the user that would not be available through other means.

B. System Design

Figure 25 shows how the different system components interact.

The user performs the user actions described in the previous section: looking by moving the eyes, and covertly attending without looking at the target directly. The user also interacts directly with the game through the keyboard to move around in the virtual environment.

EEG is measured in order to detect covert attention. EOG is measured to detect eye movement. This is done using Biosemi ActiveTwo hardware with 32 active electrodes, 7 additional flat-type electrodes on the skin, and separate CMS plus DRL. The raw data is sent over USB to the computer, where the Biosemi ActiView software sends the data over TCP/IP to the signal analysis software.

SnakeStream handles reading data from ActiView, passing the data in the appropriate formats to the signal analysis pipelines, and sending the prediction results from the pipelines on to the game environment of Wild Photoshoot. Snakestream works together well with the Golem and Psychic Python libraries, and supports the use of different markers and different sliding windows for each pipeline.

Within the game, keyboard input is used to move around the virtual world, eye movement to adjust the camera angle, and covert attention to shoot a great picture of the animal. The game can send markers to the EEG stream to give commands to the signal analysis software,

and to annotate the data for later offline analysis of the recordings. Because of limitations of the game engine software, it has to use the marker server to do this.

The marker server is a small application that receives marker values over TCP/IP and forwards them to the parallel port so it is added to the EEG stream. It also implements a simple queuing mechanism to ensure that markers do not get overwritten.

V. Online Evaluation

Due to time constraints, the online evaluation has not yet been executed. However, the preparations are ready. The online evaluation will look into two aspects. One is the influence of the online immersive situation on the signals measured and the classification performance for covert attention. The second is to evaluate the system usability and user experience when adding the eye movement camera adjustment to the interaction.

The goal is to run the main experiment for 10 participants. Each session will start with a recording of a clinical session, so we know for each participant what the theoretical performance is. This can then later be compared to the performance on the in-game training sessions. Next there will be two game sessions: one with eye movement and covert attention, and one with just covert attention where the camera is only adjusted by mouse. This way we can determine the effect of the eye movement based camera adjustment. After each game session there will be questionnaires to evaluate usability and user experience. The experimental design will be of a random crossover type, in which participants are randomly assigned to either the group where first only covert attention is used and secondly also eye movement or the group for which the order is inversed. This way possible learning or conditioning effects will be averaged out.

We will use two different questionnaires, administered to the participants in such a way that they only have to fill in one page with questions. The two different questionnaires are the SUS [14] and an adapted version of the presence questionnaire by Witmer et al. [15]. The SUS provides us with a standardized well validated scale that has been tested extensively in the field of HCI. For a more in-depth knowledge of the presence, immersion and control the user experiences within our game, we took the most interesting scales from the presence questionnaire and added some items particularly of interest to BCI research. Whereas normal input devices such as the mouse and keyboard provide the system with reliable input, using a BCI will not provide the user with perfect transmission of their intention to the system. This has its reflection on the user and we want to measure to what extent it alters the user experience.

VI. Discussion and Conclusions

The goal of this project was to develop a prototype that uses naturally occurring neurophysiological activity for natural user tasks, applying them in a way that supports intuitive interaction, with natural system responses. Pipelines for overt and covert attention have been developed and evaluated. A game that uses them in an intuitive manner has been designed and implemented, as well as a platform that provides the communication glue between each of these components.

Covert attention into four directions is detectable, but not well enough to be used as such in a game. The current game therefore only uses two classes: left and right. Detection accuracy did not decrease significantly for different fixation points. Around 80 trials will be enough for a training set for two classes. Larger trial windows result in higher performances, but this has not been tested beyond 1.5 seconds.

Horizontal eye movement seems to be detectable quite well. Vertical eye movement seems a little bit more problematic: sometimes it does not show even though it is expected. This could be an inherent problem as the vertical distance between targets is smaller than the horizontal distance. Applying eye blink detection and correction could also improve performance. Optimal window length and training protocol still need to be determined.

However, most of these results are based on relatively little data. Analysis should be redone at a later stage, including all datasets. Possibly additional sets need to be recorded.

Other future work consists of performing the online experiments, perhaps implementing the CA3 pipeline that worked better than the others, using eyes closed data to determine a personalized alpha band, and evaluating eye movement detection based on EEG without the need for separate EOG electrodes.

References

[1] B. Allison, B. Graimann, and A. Gräser, "Why Use A BCI If You Are Healthy?" in *BRAINPLAY 07 Brain-Computer Interfaces and Games Workshop at ACE (Advances in Computer Entertainment) 2007*, 2007, p. 7.

[2] R. Jacob and K. Karn, *Eye tracking in human-computer interaction and usability research: Ready to deliver the promises (Section Commentary)*. Elsevier Science, 2003.

[3] M. Posner, "Orienting of attention," *The Quarterly Journal of Experimental Psychology*, vol. 32, no. 1, pp. 3–25, 1980.

[4] R. Wright and L. Ward, *Orienting of attention*. Oxford University Press, USA, 2008.

[5] M. I. Posner and S. E. Petersen, "The attention system of the human brain," vol. 13, pp. 25–42, 1990.

[6] M. Worden, J. Foxe, N. Wang, and G. Simpson, "Anticipatory biasing of visuospatial attention indexed by retinotopically specific alpha-band electroencephalography increases over occipital cortex," *Journal of Neuroscience*, vol. 20, no. 6, p. 63, 2000.

[7] S. P. Kelly, E. Lalor, R. B. Reilly, and J. J. Foxe, "Independent brain computer interface control using visual spatial attention-dependent modulations of parieto–occipital alpha," 2005, pp. 667–670.

[8] M. A. J. van Gerven, A. Bahramisharif, T. Heskes, and O. Jensen, "Selecting features for BCI control based on a covert spatial attention paradigm," vol. doi:10.1016/j.neunet.2009.06.004, 2009.

[9] T. A. Rihs, C. M. Michel, and G. Thut, "Mechanisms of selective inhibition in visual spatial attention are indexed by α-band EEG synchronization," vol. 25, no. 2, pp. 603–610, 2007.

[10] M. A. J. van Gerven and O. Jensen, "Attention modulations of posterior alpha as a control signal for two-dimensional brain-computer interfaces," vol. 179, no. 1, pp. 78–84, 2009.

[11] A. Bahramisharif, M. A. J. van Gerven, T. Heskes, and O. Jensen, "Covert attention allows for continuous control of brain-computer interfaces," vol. 31, no. 8, pp. 1501–1508, 2010.

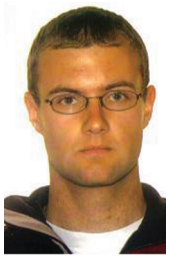
[12] A. Bulling, D. Roggen, and G. Tröster, "Wearable eog goggles: eye-based interaction in everyday environments," in *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2009, pp. 3259–3264.

[13] R. Barea, L. Boquete, M. Mazo, and E. López, "Wheelchair guidance strategies using eog," *J. Intell. Robotics Syst.*, vol. 34, no. 3, pp. 279–299, 2002.

[14] J. Brooke, *SUS: a "quick and dirty" usability scale.* London: Taylor and Francis, 1996.

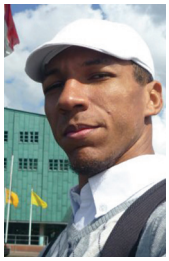
[15] B. G. Witmer and M. J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence: Teleoper. Virtual Environ.*, vol. 7, no. 3, pp. 225–240, 1998.

Danny Plass-Oude Bos did her internship at the University of Nijmegen, implementing physiological artifact detection in an online EEG-based BCI system. In 2008 she obtained her master in Human-Computer Interaction on BrainBasher, looking into the user experience of using BCI for games. At the moment she is working as a PhD student at the University of Twente, still attempting to merge BCI with HCI by researching how BCI can be made a more intuitive means of interaction.



Matthieu Duvinage , TIME student, holds Electrical Engineering degrees from the Facult Polytechnique of Mons (UMons, Belgium, 2009), and Electrical Engineering degree from SUPELEC (France, 2009), a degree of fundamental and applied physics from Paris Sud XI Orsay (France, 2009) and a degree of management science from the School of Management at the University of Louvain (UCLouvain, 2011). His master thesis at Multitel (Mons, Belgium) dealt with robust low complexity speech recognition. He obtained an F.R.S-FNRS grant for pursuing a PhD thesis about developing a lower limb prosthesis driven by a neural command in close partnership with the Free University of Brussels (ULB).

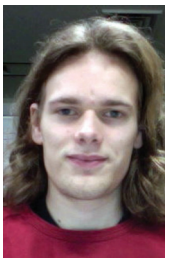
Oytun Oktay obtained his bachelors degree in electronics engineering from Gebze Institute of Technology in 2008 with a insight of numerical electromagnetics. He started computational science and engineering master's program in Istanbul Technical University but left it after two years for a research assistant position at Namik Kemal University. Now he is working as a research assistant in Electronics and Communication Engineering Department on the subject of functional brain networks.



Jaime Delgado Saa obtained his Bachelor and Master in Electronics Engineering in 2003 and 2008 respectively, at the Universidad del Norte (Colombia). From 2004 to 2009 he worked at the Universidad del Norte at the department of Electrical and Electronics Engineering. Currently he is a PhD student at Sabanci University at the Vision and Pattern Analysis group, working in the field of Brain Computer Interfaces. His work is focuses in the analysis temporal and spatial of EEG signals.

Huseyin Gürüler works as a lecturer at Mugla University in the department of Electronics and Computer Science. He has a bachelor degree in the field of Electronics and Computer from Marmara University, Istanbul, and master degree in the field of Statistics and Computer, Mugla University, Mugla, Turkey. His MSc thesis is about data mining and knowledge discovery in student databases. Now, he is a PhD student, researching RLS disease diagnosis from physiological signals.

Dr. Ayhan Istanbullu received his undergraduate degree (1993) and his PhD from the Electronic and Computer Science Education Department of the Gazi University (2003), Turkey He worked as an instructor at the University of Mugla, Turkey in the Department of Electronic and Computer Science Education (2001-2006). He currently works as assistant professor in the Computer Engineering Department of Balikesir University, Turkey. He has participated in the European Remote Radio Laboratory project (EU LdV). His current research interests include the investigation of information technologies to support electronic and computer engineering education, mobile learning and intelligent tutoring.



Marijn van Vliet has recently graduated from Twente University in the field of Human Media Interaction, where he devoted most of his time studying Brain-Computer Interfaces. He did an internship in Tokyo on virtual worlds and is recently accepted for a PhD position at the University of Leuven, where he will continue his BCI work.

Bram van de Laar obtained his Bachelor degree in Computer Science (2006) and Master degree in Human Media Interaction (2009). With a broad interest in technology, such as: 3D, video, networking, music, sounds, haptics, brain-computer interfacing and physical exertion, Bram tries to combine different media to create a synergy by exploiting different modalities. User experience and 'added value' play an important role in this philosophy. As a PhD candidate Bram gets the space to explore the possibilities in this area.

Mannes Poel is a senior reseacher at the Human Media Interaction group of the department of Computer Science, University of Twente. He has a background in mathematics and theoretical computer science. After working on verification of concurrent processes his interest shifted towards Neural Networks, Machine Learning and Artificial Intelligence. At the moment his research focuses on Machine Learning in the context of Human Behavior Computing, including BCI. Currently he supervises several PhD students in these areas. He (co-)authored several papers on Machine Learning in HCI and BCI, some of them also focusing on affective dialogue management.

Linsey Roijendijk obtained her Bachelor degree in Computer Science (2008) and her Master degree in Artificial Intelligence specialized in cognitive research (2009). Since November 2009, she is working as a PhD student at the Biophysics department of the Donders Institute for Brain, Cognition and Behaviour, Nijmegen, the Netherlands. Currently, her work focuses on investigating variabilities in EEG based Brain Computer Interfaces (BCI) and using covert spatial attention for BCI.

Luca Tonin obtained his Master degree in Electronic Engineering (2008) at University of Padova. Integration between Brain-Computer Interface and robotic devices was the main topic of his master thesis. Since January 2010, he is working as PhD student at EPFL in the CNBI (Chair on Brain-Machine Interface) laboratory. He is actively involved in TOBI (Tools for Brain-Computer Interface) an European Project started in November 2008. Currently, his research and PhD thesis are focused on covert attention for BCI purposes.

Ali Bahramisharif got his bachelor's and master's degree in the field of Electrical and Electronics Engineering from Sharif University of Technology and Tarbiat Modarres University, Tehran, Iran, respectively. Since 2008, he is working as a PhD student at the Intelligent Systems group of Radboud University Nijmegen, and as of February 2009 he has appointed as a part-time researcher at the Donders Institute for Brain, Cognition and Behaviour, Nijmegen, the Netherlands. Research interest includes brain-computer interfaces (BCI), machine learning methods and source localization. His recent research focuses on different aspects of covert spatial attention.

Boris Reuderink obtained his Masters degree at the University of Twente in 2007, after working on different machine learning problems. He combines his big interests brains and intelligence with his PhD position at the University of Twente, in which he focuses on making brain-computer interfaces work in real-world settings for healthy users.

Continuous Interaction with a Virtual Human

Dennis Reidsma, Khiet Truong, Herwin van Welbergen, Daniel Neiberg, Sathish Pammi, Iwan de Kok, and Bart van Straalen

***Abstract*—Attentive Speaking and Active Listening require that a Virtual Human be capable of *simultaneous perception/interpretation and production* of communicative behavior. A Virtual Human should be able to signal its attitude and attention while it is listening to its interaction partner, and be able to attend to its interaction partner while it is speaking – and modify its communicative behavior on-the-fly based on what it perceives from its partner. This report presents the results of a four week summer project that was part of eNTERFACE'10. The project resulted in progress on several aspects of continuous interaction such as scheduling and interrupting multimodal behavior, automatic classification of listener responses, generation of response eliciting behavior, and models for appropriate reactions to listener responses. A pilot user study was conducted with ten participants. In addition, the project yielded a number of deliverables that are released for public access.**

***Index Terms*—Virtual Humans, Attentive Speaking, Listener Responses, Continuous Interaction**

I. Introduction

Continuous Interaction is one of the fundaments underlying Attentive Speaking and Active Listening for Virtual Humans (VHs). Attentive Speaking and Active Listening require that a Virtual Human be capable of *simultaneous perception/interpretation and production* of communicative behavior. A Virtual Human should be able to signal its attitude and attention while it is listening to its interaction partner, and be able to attend to its interaction partner while it is speaking – and modify its communicative behavior on-the-fly based on what it perceives from its partner. This report presents the results of a four week summer project that was part of eNTERFACE'10. The project resulted in progress on several aspects of continuous interaction such as flexible and adaptive scheduling and planning including graceful interuption, automatic classification of listener responses, generation of response eliciting behavior, and models for appropriate reactions to listener responses. We made a start on evaluating the results in classification experiments as well as in a pilot user study. In addition, the project yielded a number of deliverables that are released for public access, among which a public release of Elckerlyc, a new platform for building Virtual Humans, and a database of motion capture animations containing over 100 direction-giving-task related gestures in the route giving domain.

II. Background and Motivation

The design of VHs often focuses on the combination of speech with gestures in conversational settings. They tend to be developed using a turn-based interaction paradigm in which the user and the system take turns to talk. If the interaction capabilities of VHs are to become more human-like and VHs are to function in social settings, their design should shift from this turn-based paradigm to one of continuous interaction in which all partners perceive each other, express themselves, and coordinate their behavior to each other, continually and in parallel [1], [2]. This requires the realizer to be capable of immediate adaptation – in content and in timing – to the dynamics of the environment and the user.

The main objective of this project is to explore this kind of coordination behavior in ECAs, modeling and implementing the sensing, interaction and generation for what we call continuous interaction. A continuous interactive ECA will be able to perceive the user and generate conversational behavior fully in parallel, and can coordinate behavior to perception continuously – a capability which is not yet present in most state-of-the-art ECAs.

One of the major sources of overlap in conversation, and therefore a very good domain for addressing continuous interaction capabilities in Virtual Humans, are Listener Responses [3]. We will take a first step towards the global goal by making a VH that is capable of actively dealing with Listener Responses from the user, while the VH is speaking.

This research has kindly been supported by the GATE project, funded by the Dutch Organization for Scientific Research (NWO) and the Dutch ICT Regie, and by the FP7 NoE SSPNet

A. Structure of this Report

This report is structured as follows. Section III gives an introduction to the theoretical background of Responses and Attentive Speaking on which we based our approach. Section IV presents the overall system setup of an interactive Virtual Human system as we used it in our development and experiments. Sections V and VI introduce the corpora that we used, and analyse them with respect to the characteristics of Responses that we find in them. Section VII is dedicated to the automatic classification of Responses. Sections VIII and IX concern behavior scheduling and planning for continuous interaction for Virtual Humans: they describe the already existing possibilities as well as the new developments achieved in this project. Sections X and XI discuss our work on the Response Elicitation pilot user study. The paper ends with a discussion of what we have achieved, and where we need to go next.

III. Listener Responses and Attentive Speaking

An active listener shows his or her interest, attention and/or attitude with respect to the speakers utterances in many ways: through gaze direction and eye contact, using face expressions, using short utterances like "yeah", "okay", and "hm-m", etcetera. An attentive speaker will give the listener opportunities for such responses, but will also actively receive the responses, and adjust his or her utterances to the occurrence and content of these responses. In this section, we discuss (listener) responses and attentive speaking in more detail.

A. Responses and Listener Responses

The conversational context is that of a VH is explaining a certain route on a map to the user. This conversational context implies that the VH is mostly speaking (is a Speaker), and the user is listening (is a Listener). At some point, the user starts to talk. This may be to give feedback or it may be a question, answer, statement, or other full contribution to the conversation. The user's utterance may overlap an utterance of the VH, or it may be at a moment that the VH was silent.

We refer to as everything the Listener says as "Responses", which implies the role in the conversation.

The Listener commonly utter responses such as "yeah", "mhm", "uhu". Fujimoto [3] propose to call these short utterances Listener Responses. These are short utterances or vocalizations which are interjected into the Speakers' account without causing an interruption, or being perceived as competitive of the floor. They serve many functions, were the most important is to signal that the Listener

hears that the Speaker is talking and nothing more than this neutral function. This function is sometime called back-channeling and is not mandatory. Another common function is signaling understanding to what the speaker is saying. This function is commonly referred to as Acknowledgment. In addition, they may be used as carriers of more subtle information, conveyed by intonation, voice quality, face expression, rhythm, content of the words, etcetera.

From a more generalized point of view, a Response may convey information regarding Understanding (whether the Listener understands the utterance of the Speaker), Attentiveness (whether the Listener is attentive to the speech of the Speaker), Attitude [4] and Affect [5], and may be described as being competitive (interruptive) or cooperative (non-interruptive) [6].

B. Attentive Speaking

A good speaker pays attention to the listener. He moderates his speech and tailors it to the responses from the listener. Listeners are not merely listening, but are co-narrating along with the speaker [7]. A good virtual human should be able to do this as well.

This interaction between speaker and listener works in various ways. To illustrate this we will give a few examples from literature. Clark and Krych [8] identify several strategies in dialogue that depend on opportunities that arise, intentionally or not, mid-sentence. They claim that speakers make the alterations instantly, typically initiating them within half a second of the opportunities becoming available.

One of the strategies the speakers apply to coordinate their speech is self-interruption. If the listener provides a response in mid-utterance which makes another utterance more relevant at the time (for instance, because the listener signals non-understanding and an elaboration is needed), the speaker cuts of his utterance and starts a new one (see Example 1).

Interaction Example 1 Self-interruption.

Speaker: So starting from the square, you go...
Listener: euhm?
Speaker: I mean the square with the obelisk on it.

The observations from Goodwin [9] work on a lower level. In his observation, the speaker does not change what he says based on the responses from the listener, but the timing is coordinated with the listener. He makes a distinction between continuers and assessments. Continuers simply acknowledge the receipt of the talk just heard and signals the speaker to continue his talk. Assessments are the result of an analysis of the speakers' talk by the listener based on which, the listener has produced an action that is responsive to the particulars of the talk. Continuers are usually placed between two subsequent speech units, while assessments are placed in the midst of a unit and completed before a new unit starts. This is actually facilitated by the speaker. So, if the speaker recognizes an assessment and is about to start a new unit, he delays this unit (e.g. by an inhalation or production of a filler) until the listener has completed his assessment.

This coordination does not only facilitate vocal responses from the listener, also nonverbal signals are dealt with by the speaker. Goodwin [10] showed that speakers are highly sensitive to listeners gaze. If they start a sentence and discover the listener is not looking at them, they restart (and often rephrase) when the listener look back.

This is merely a selection of situations and strategies in which the speaker moderates his speech to the responses of the listener. There are many more, which we did not cover, but they illustrate the type of coordination we are aiming to achieve with our system. It is our aim to create a system which is technically able to achieve the same level of continuous interaction with the user as illustrated by these examples.

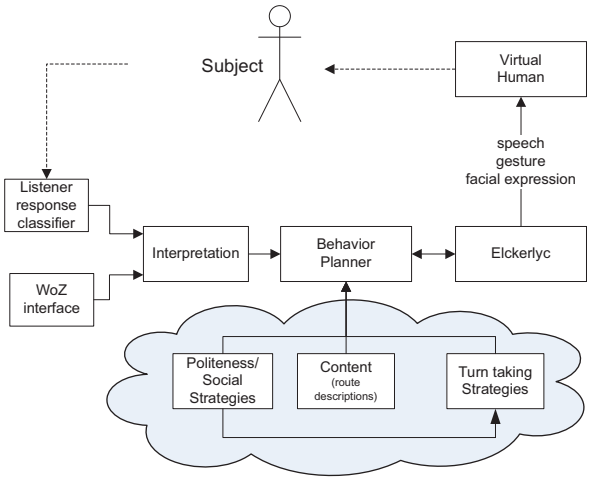


Fig. 1: System architecture

IV. System Overview

Fig. 1 gives an overview of the architecture of the interactive virtual human system that we have developed. The virtual human explains a route through a city, in such a way as to elicit Responses form the user. We detect the occurrence of Responses(e.g., "uh-huh", "mmm") using both non-verbal vocalization analysis and a Wizard of Oz interface. The behavior planner specifies the behavior to be realized on the basis of politeness and social strategies and conversation content (a specification of the route to explain). The behavior is constructed using speech, gestures, gaze, and face expressions.

If Responses occur, Elckerlyc is instructed to gracefully interrupt the currently running behavior or to retime or re-parameterize (speak louder, increase the amplitude of gestures etc.) its behavior. New behavior can be constructed by selecting and inserting new BML fragments in order to react to interruption. The exact method of feedback handling is influenced by turn-taking strategies and politeness/social strategies. The different components are connected using the SEMAINE framework [11].

V. Corpora

We used two corpora in this project, namely the MapTask corpus [12] and the MultiLis corpus [13]. These corpora were used for two purposes: (1) to find out more about the content and timing of listener responses, and (2) as training and testing material for our classifiers.

A. The MapTask Corpus

The HCRC Map Task Corpus is a set of 128 dialogues. The task is for one subject to explain a route to another subject. The one who explains the route is denoted as the "giver" and the one who receives the explanation as the "follower". Half of the dialogs were recorded under a face-to-face condition and the other half under a non-visible condition. We used the dialogues from the face-to-face condition since it is closer to our scenario of an interaction with a Virtual Human. The two conversations labeled as `q3ec1` and `q3ec5` where discarded due to a buzz in the speech signal.

The segmentation of the dialog in the MapTask corpus is based on manual annotations. For the analyses and experiments discussed in this report, we chose to use instead segmentations based on an ideal voice activity detector, because that will more closely reflect the conditions that we will encounter in the application of a conversation with a Virtual Human. We segment the corpus into *talk spurts* [14], defined as a minimum voice activity duration of 50ms separated by a

TABLE I: Confusion matrix for the annotation of overlapping talk spurts on Competitiveness. Cohen's κ=0.60; Krippendorff's α (nominal) = 0.45.

	COMPETITIVE	COOPERATIVE
COMPETITIVE	88	77
COOPERATIVE	40	319

minimum inter-pause of 200 ms. These talk spurts are referred to as "ideal VA Detector talk spurts". If a talk spurt is comprised of more than one MapTask segment, the talk spurt is labeled with the label from the first MapTask segment included in the talk spurt. This gives a consistent segmentation strategy, uses all relevant speech, and the results will better resemble the condition when a real voice activity detector is used.

To simulate real-world conditions even closer, we additionally created a second set of talk spurts using the OpenSmile voice activity detector. For each ideal VA Detector talk spurt, 3 seconds of silence is added in front, and 10 seconds of the original audio following the ideal VA Detector talk spurt is added to the end. Then the first talk spurt detected by the OpenSmile voice activity detector, configured with minimum voice activity duration of 100 ms and a minimum inter-pause of 200 ms, is assigned the same label as the "ideal VA Detector talk spurt" and saved for further experiments. If no talk spurt is detected, then the corresponding label is thrown away. We refer to these segments as "OpenSmile VADetector talk spurts".

We used the official MapTask annotations concerning the distinction between Acknowledgement Moves (MTACK) and other talk spurts (NONMTACK). The precise definition of an Acknowledgment Move is found in [15], but they closely resemble the term Listener Response [3] and thus serve our purpose. According to Carletta et al. [15], these MapTask annotations are good ($\kappa = .83$), although one of the largest confusions did involve the Acknowledgement Moves (confusion with Ready and Reply-Y).

In addition, we annotated part of the data with information whether the talk spurt intends to take the floor (COMPETITIVE) or not (COOPERATIVE).

The following talk spurts were annotated:

- We only annotated NONMTACKs, as MTACKs are supposed to be COOPERATIVE by definition.
- We annotated only Responses in overlap (Listener's talk spurt starts between the start and the end of the Speaker's talk spurt) because the COOPERATIVE/COMPETETIVE dimension only makes sense for overlapping talk spurts.
- We only annotated NONMTACKs, which does not have any MTACKs within the local overlap. For example, a NONMTACK which is intercepted in overlap by MTACK is excluded.

In the data that we used, there are 1232 candidate talk spurts to be annotated. Of these, 524 talk spurts (quad 1-4) were labelled by two annotators. The confusion table and relibaility values are given in Table I. The level of agreement for this annotation is in the range of highly subjective annotations [16]; the annotators agree on a certain amount of talk spurts being COOPERATIVE, but have difficulty agreeing on which talk spurts are COMPETITIVE.

B. The MultiLis Corpus

Because the mapTask corpus does not contain video recordings, it could not provide us information about nonverbal responses and nonverbal respose elicitation behavior such as gaze, nods, and face expressions. For this, we used the MultiLis corpus.

TABLE II: Top 20 most frequently occurring Acknowledgement talk spurts in the MapTask corpus (MTACK talk spurts), accounting for 7313 out of 9823 of these talk spurts.

count	word	count	word	count	word	count	word
2773	right	264	oh	93	got	66	a
1459	okay	227	the	89	it	65	to
525	mmhmm	153	that's	86	you	63	fine
521	uh-huh	145	no	82	that	58	i've
380	yeah	133	i	73	mm	58	aye

The MultiLis corpus [13] is a Dutch spoken multimodal corpus of 32 mediated face-to-face interactions totalling 131 minutes. Participants were assigned the role of either speaker or listener during an interaction. The speakers summarized a video they have just seen or reproduced a recipe they have just studied for 10 minutes. Listeners were instructed to memorize as much as possible about what the speaker was telling. In each session four participants were invited to record four interactions. Each participant was once speaker and three times listener. What is unique about this corpus is the fact that it contains recordings of three individual listeners to the same speaker in parallel, while each of the listeners believed to be the sole listener. The speakers saw one of the listeners, believing that they had a one-on-one conversation. The aim of the corpus was to collect responses from different individuals to the same speaker context. The corpus illustrates the individual differences in listening behavior, but also includes differences in the amount of responses that individual speakers were able to elicit.

VI. ANALYSIS OF RESPONSES IN HUMAN-HUMAN INTERACTION

This section provides an analysis of properties of Responses from the MapTask corpus. Rather than providing a complete analysis, we only adress the parts which are crucial for the design of the system. Table II shows the most frequently occurring word content for MTACK talk spurts, accounting for 7313 out of 9823 of these talk spurts.

A. MTACK *Content*

Figure 2 shows the duration of MTACKs vs. the other dialog moves. It is clear that MTACKs have a short duration and may (partially) be detected by duration alone. Concerning overlapping speech, we can observe the following: The proportion of overlapped speech in the MapTask corpus is 9.1%, the proportion of MTACKs is 7.3% and the proportion of MTACKs in overlapped speech is 34.9%. Thus, MTACKs are more common in overlap than in non-overlapped speech.

B. *Gaps Following* MTACK *talk spurts*

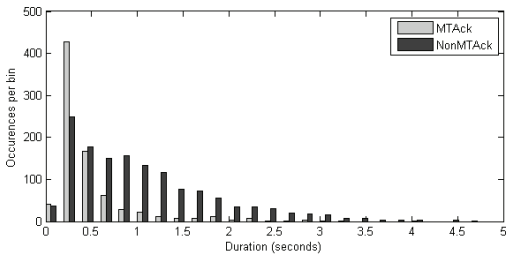


Fig. 2: Duration of MTACKs vs. duration of other dialog moves, using bins of 200 msec.

Since we are trying to build a Virtual Human that can deal with Responses in a continuous interactive way, we also investigated the continuation talk spurt of the Speaker following the onset of a MTACK Response. For all MTACK Responses that do *not* interrupt the Speaker (i.e. the Speaker continues speaking afther the onset of the Response) we calculated the gap between the *end* of the Response and the *beginning* of the continuation talk spurt of the speaker. This gap has a negative value if the Speaker continues speaking before the end of the Response. Figure 3 shows the distribution of the gap for all Speaker continuation talk spurts. The figure shows that the Speaker commonly continues to speak after roughly 0-400 ms. It also shows that negative gap – that is, overlap – is not uncommon. This means that for a responsive dialog with a Virtual Human, Responses from the user need to be classified before they are finished. This might be done using a speech recognizer running in incremental mode or by using a specialized detector. Since a speech recognizer will only detect lexical content, the special prosodic characteristics of listener responses cannot be accounted for. It is also an open question how well a speech recognizer will perform in detecting grunt-like nature of some listener responses. This is because Responses such as "mmhmm" are tokens which are shown to be unstable in their allophonic surface realizationsm, and there is no standardized annotation scheme for these [17].

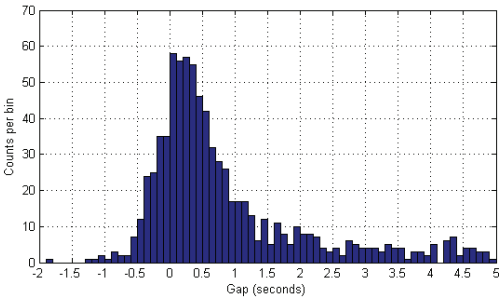


Fig. 3: The gap or overlap (negative gap) between a MTACK Response and the incterlocutors' continuation using bins of 100 ms.

C. Duration of COMPETITIVE and COOPERATIVE Responses

Figures 4 and 5 give the distribution of the duration of COMPETITIVE and COOPERATIVE Responses, and of the durations of the *overlap* for both types of Responses.

We notice that these distributions are different. Short overlaps around 100 ms are more likely for cooperative speech rather than for competitive speech. The most likely overlap duration for cooperative speech is around 100ms, and it wears off around 2100 ms. The most likely overlap duration for competitive speech is around 300ms, and it wears off around 1100 ms. This means that a detector should give a decision as early as possible after the onset of the Response: preferably at 300ms, but no later than 1100ms.

Secondly, we observe that cooperative talk spurt tend to be shorter in durations than talk spurt for competitive speech. This means that duration may be used as a feature for competitiveness, but still the decision to stop talking when incoming speech are observed in overlap, is constrained by the observed durations of overlap explained in the previous paragraph. Thus, there is a trade-off between these two constraints, the different durations of talk spurt and overlap.

VII. CLASSIFICATION OF LISTENER RESPONSES

This section deals with the classification of Responses based on audio input. Being an attentive speaker includes giving attention to

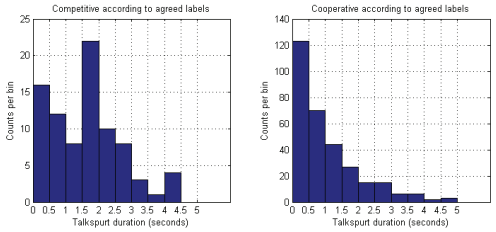


Fig. 4: Durations of talkspurts in overlap with no MTACK context (within the overlap). To the left are COMPETITIVE and to the right COOPERATIVE Responses.

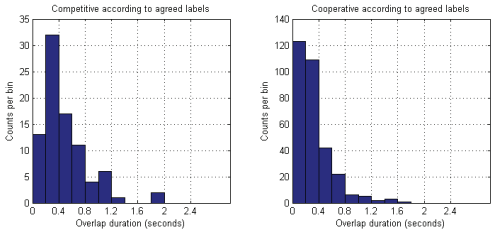


Fig. 5: Durations of the overlap with no MTACK context (within the overlap). To the left are COMPETITIVE and to the right COOPERATIVE Responses.

what the listener says and taking appropriate action. First of all, this involves recognizing Responses and the information they convey. We approach this by classification of incoming voice activity in the audio channel. As mentioned before (Sections IV and VI), it is important to classify incoming talk-spurts before they end, preferably within 300-700 msec of the onset of the speech.

The classifiers are needed for the system to determine, given incoming speech from the user, what the reaction of the Virtual Human should be. If the incoming speech overlaps speech from the Virtual Human, the decision may be to stop speaking, or to continue speaking in overlap. The latter makes sense when the incoming speech is a COOPERATIVE Response. If the incoming speech does not overlap, the reaction of the Virtual Human should very much be determined by the information conveyed by the Response. For example, an MTACK Response probably requires no change of the dialog plan; a Response expressing non-understanding or disagreement may require elaboration, initiation of a clarification dialog, or other more drastic revisions of the dialog plan. The last type of Responses are not dealt with by the classifiers presented here.

We classify Responses using the cascade shown in Figure 6. The first classifier in the cascade is trained on the MapTask corpus to distinguish MTACK talk spurts from other talk spurts. MTACK talk spurts are, among other things, by definition COOPERATIVE Responses. Talk spurts *not* classified as MTACK may be COOPERATIVE or COMPETITIVE (see Section V-A). Concerning these NONMTACK talk spurts we focus on talk spurts produced by the user in overlap as they more urgently require a decision from the Virtual Human (namely, to continue speaking even while the user is speaking too, or not). We tried two different approaches to classify those talk spurts. The first approach was based on classifying them according to the theoretical distinction between COOPERATIVE and COMPETITIVE Responses. The second approach was pragmatically oriented, based on *predicting the outcome of the overlap*, that is, predict whether the Speaker or the Listener is the one who continues speaking after the overlap. The third approach is a hybrid approach, and attempts to exploit a possible relation between the pragmatic "outcome of overlap" rules and the theoretical distinction from the first approach.

All classification experiments were performed using openSMILE [18] for automatic feature extraction and *libsvm* [19] for classification.

In summary, this leads to four main classification tasks.

- **Classifier I** Classification of all Responses into MTACK / NONMTACK
- **Classifier IIa** Classification of NONMTACK, produced in overlap, into COOPERATIVE / COMPETITIVE, based on our manual annotations (the theoretical approach)
- **Classifier IIb** Prediction of the outcome of the overlap for all NONMTACK produced in overlap (the pragmatic approach)
- **Classifier IIc** Classification of NONMTACK, produced in overlap, into COOPERATIVE / COMPETITIVE, based on the hybrid approach

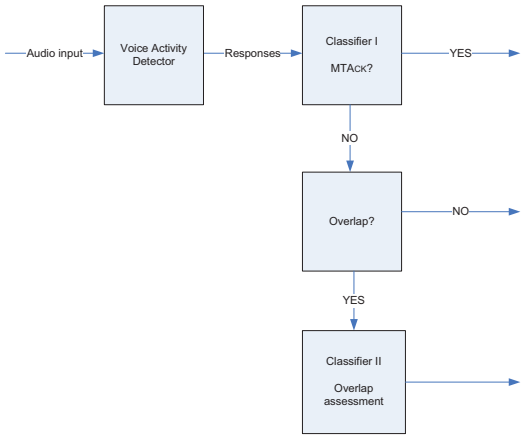


Fig. 6: Cascade used to classify incoming Responses from the user.

A. Maximum latency classification

The analysis of the gap after a listener response in Figure 3, showed the presence of a negative gap, i.e. an overlap. This means a decision whether incoming speech is a listener response or not has to be made before the the listener response ends. Thus, we consider a maximum latency design for the detector. It is implemented as a voice activity detector which sends an end message after the talk-spurt ends, or at a predefined duration threshold, denoted as the maximum latency. If the duration reaches the threshold, it continues to work as normal voice activity detector internally, otherwise it might trigger again. Note that the detector may trigger before the maximum latency if the talkspurt is shorter than the threshold subtracted by the minimum inter pause threshold. For online detection, this maximum latency design was implemented in openSMILE [18].

B. Feature trajectories as length-invariant Discrete Cosine Coefficients

To parameterize the trajectories of each feature through out a talkspurt, we use DCT coefficients invariant to segment length:

$$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}(n+\frac{1}{2})k\right) \qquad k = 0,\ldots,N \qquad (1)$$

where N is the segment length, x_n is the feature value at time n and X_k is the k'th coefficient.

These DCT coefficients are much faster to compute than polynomial regression coefficients, since polynomial regression require matrix inversion. This makes length invariant DCT coefficients more suitable for online systems. The 0'th coefficient is equal to the arithmetic average, which means if it is omitted, then only the relative shape of a trajectory is parametrized. This property is useful for parameterizing features which has an highly speaker dependent additive bias, such as F0. These DCT coefficients has been used to visualize a single average trajectory of multiple speech segments [20]. When a DCT is applied on MFCCs, one obtain the cepstrum modulation spectrum. The usage of length-invariant cepstrum modulation spectrum was first introduced by [21], although no specific term was used at the time. The cepstrum modulation spectrum has been use for speech recognition [22] and in its length invariant version for affective detection [23]. By omitting the 0th DCT coefficient for MFCCs in the time dimension, then any channel mismatch which appear as an additive bias in the quefrency will not cause any problem. Our experiments will determine whether omitting the 0'th coefficient still gives a decent classifier. Unless anything else is stated, the 0'th DCT coefficient in time dimension is always omitted.

	TRAINING	DEVELOPMENT	EVALUATION
MTACK	775	482	537
NONMTACK	1315	677	1138

TABLE III: Number of talkspurts used for training, developing and testing Classifier I.

C. Support Vector Machine classification

All classifiers use Support Vector Machines (SVM) with Radial Base Function Kernel as implemented in *libsvm* [19]. In a few cases, we consider a minor but pragmatic modification to the standard SVM scheme, which is here denoted as *rescaling*. When feature sets of different nature are evaluated on the development set, quite different optimal γ values are found for each feature set. The γ parameter in a radial base kernel is proportional to the inverse of the variance in a Gaussian. This means that if each feature set would have different γ, then a more optimal decision hyperplane may be found. One solution to this problem uses multiple kernels [24]. Here we offer a simple and pragmatic solution for this problem. After each feature set f has been evaluated separately, the optimal $\gamma^f_{optimal}$ is saved. When the combined feature set is created, a rescaling procedure is applied, after the regular scaling to $[-1,1]$ or N(0,1) . The original scaled feature set x^f for each feature set is then rescaled by

$$\hat{x}^f = \frac{\gamma^f_{optimal}}{\min_{i=1\ldots I}\gamma_i} \qquad (2)$$

where i denotes the indexes for all γ in the grid search. This rescaling procedure can be applied to most standard SVM implementations with only minor modifications.

D. Experimental setup

For all experiments, the training set consists of so-called quads 1-4, the development set holds quads 5-6 and the evaluation set holds quads 7-8. The number of talkspurts used in the classification experiments can be found in Table III. The SVM regularization parameters are optimized on the development set, and the best parameters are then used for test on the evaluation set.

As explained in Section V-A, the first series of experiments explores features and combinations thereof under the assumption that an ideal voice activity detector is available (referred to as the "ideal VAD talk spurt" situation). In the second series of experiments, the ideal segmentation is replaced by an actual voice activity detector

based on energy thresholds (referred to as the 'OpenSmile VAD talk spurt' situation). This is done to ensure that the classification results reflects real life conditions as closely as possible. Since a parametrization of the trajectory of each feature is used, the resulting models are expected to be sensitive to mismatch in segmentation. Thus, the same segmentation should be used for training and on-line recognition. Still we considered safe to extrapolate the results from the first series, and use the best found combinations of features for the experiments using the 'OpenSmile VAD talk spurts'.

E. Classifier I: MTACK *vs. Other*

1) Features: For the task of classifying incoming speech as a MTACK or not, a set of acoustic features are considered.

- F0: Back-channels has been shown to have a rise or drop in F0 [25][20].
- Intensity: Back-channels has been shown to have distinct intensity contours [25]
- MFCC: Similar lexical content, see Table II, may be captured by MFCCs.
- Duration: As seen in Figure 2, MTACKs have shorter duration then other type of speech. For training, the full talk-spurt duration was used, for testing, the duration up to the maximum latency threshold was used.
- Spectral Flux: Common listener responses such as "mmhmm" and "uh-huh" are relatively homogeneous throughout their realization, and spectral flux should capture this property.

All features are parametrized using DCT-coefficients 1-6 or 0-6, as described in Section VII-B. As classification method, we used a ν-SVM. The parameters g and c were optimized on the DEV set (on F_avg) through a simple gridsearch with growing sequences of the ν (sequences growing linearly) and g (sequences growing exponentially) parameters within ranges of [0.025, 0.6] and [0.0156, 4] respectively.

For this classifier, a maximum latency of 300ms or 500ms was chosen.

Feature(s)	300 ms	500 ms
F0	55	59
Intensity	60	62
MFCC with 0th	72	75
MFCC without 0th	74	75
Duration	55	71
Spectral flux	66	67
Intensity, Sp. flux, MFCC with 0th	73	76
Intensity, Sp. flux, MFCC with 0th, Dur.	75	76
Intensity, Sp. flux, MFCC without 0th	74	76
Intensity, Sp. flux, MFCC without 0th, Dur.	73	76

TABLE IV: Average F-scores in percent for "MTACK vs other" classification for all the "ideal VA Detector talk spurts" in the development set.

max latency (ms)	Features	Avg. F-score
300	Intensity, flux, mfcc without 0th	73
500	Intensity, flux, mfcc without 0th, dur	76

TABLE V: Average F-scores in percent for "MTACK vs other" classification for all the "ideal VAD talk spurts" in the evaluation set.

2) Results And Discussion: As expected, we observe in Table IV that MFCCs and duration, at least in the 500ms case, are the main contributors to the distinction between MTACK vs. NONMTACK. The combination of features did not always yield better results. However, note that we only tried a combination of features on feature-level, and that a decision-level fusion might yield better results (which will be investigated in future work). We observe that omitting the 0th DCT for MFCCs, does not hurt performance. Table V shows results for the proposed feature combinations on the evaluation set. Surprisingly little gain is achieved by using the longer maximum latency of 500 ms as compared to 300 ms. Table VI shows the results for the more realistic 'OpenSmile VAD talk spurts'. A small performance drop is observed. Furthermore, the confusion matrix in Table VII shows that it is easier to miss a LR than to miss a NON-LR.

max latency (ms)	Features	Avg. F-score
300	Intensity, flux, mfcc without 0th	68
500	Intensity, flux, mfcc without 0th, dur	69

TABLE VI: Average F-scores in percent for "MTACK vs other" classification for the 'OpenSmile VAD talk spurts' in the evaluation set.

		Classified as	
		MTACK	NONMTACK
True	MTACK	279	258
True	NONMTACK	171	967

TABLE VII: Confusion matrix of 500-Intensity-flux-mfcc-without-0th-dur, evaluated on evaluation set

F. Classifier IIa: COMPETITIVE *vs.* COOPERATIVE

This task is based on the theoretical distinction between COMPETITIVE vs. COOPERATIVE incoming speech. The classifier was trained on agreed annotations made by two human annotators who labelled a part of the HCRC Map Task Corpus on perceived COMPETITIVENESS and COOPERATIVENESS of the incoming overlapping speech (as explained in Section V-A).

1) Features: Choosing a good acoustic feature set for this task is not easy since only a few studies are available. Intensity is the most widely studied cue for interruption ([6], [26]). Speaking rate has been studied in [27] where it was noted that competitive overlappers make use of higher speaking rates. However, [28] found speaking rate to be a weak cue for competitive speech. Speaking rate is very difficult to estimate for segments lasting less than 1000 ms. Instead, we try spectral flux which has been used for estimating tempo in music [29]. While average F0 (high) has shown to be a cue for interruption (e.g., [6]), it requires adaptive estimation of F0 range and is not considered here. As shown in the analysis in Section VII-G, talkspurt duration is a good feature. Based on the experience from annotation, we noted a tension in the voice for some interruptions and competitive speech. Thus, voice quality correlates may be useful for this task. Voice quality was measured by spectral centroid, spectral kurtosis, and spectral skewness. The final set of acoustic features was comprised of:

- F0: DCT 1-6
- Intensity: DCT 1-6
- Duration: For training, the full talk spurt duration was used. For testing, the duration up to the maximum latency threshold was used.
- Spectral Flux: 0th DCT
- Voice quality: 0th DCTs of spectral centroid, spectral kurtosis and spectral skewness

TABLE VIII: Average F-scores for predicting Comp vs Coop on development set using "ideal VAD talk spurts" from the corpus

Max lat.(ms)	300	500	700	900	1100
F0	54	57	58	57	57
Int.	56	53	59	56	55
Sp. Flux	63	61	60	60	58
V.Q.	53	51	53	51	52
dur.	46	47	48	51	51
Comb1	57	52	54	55	58
Comb2	58	52	54	55	57

TABLE IX: Average F-scores for predicting Comp vs Coop on evaluation set using "ideal VAD talk spurts" from the corpus

Max lat.(ms)	300	500	700	900	1100
Sp. Flux	61	63	59	63	55
Comb1	58	54	53	54	58
Comb2	57	53	54	53	58

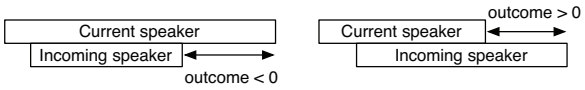


Fig. 7: The outcome of overlap that is to be predicted.

- Comb1: F0, Intensity, Spectral Flux, and Voice Quality, as specified above
- Comb2: As Comb1 with duration added, as specified above

2) Experimental setup: For training and testing the classifier, we used the COMPETITIVE and COOPERATIVE annotations that were obtained with two human annotators (see Section V-A). Only those talk spurts that were agreed upon by both annotators were included which yielded 88 and 319 talk spurts for the COMPETITIVE and COOPERATIVE class respectively. Since we have relatively little data, an N-fold-cross-validation scheme was applied for training and testing the classifier (contrary to what was done for the other classifiers). There were 4 quads available. To ensure strict separation of training, development and testing sets, in each fold, 2 quads were held out for development or testing. The models trained for optimization of the SVM parameters were trained with the other 2 quads. All possible combinations of quads with strict separation of training, development, and testing sets were made which yielded 12 folds for the optimization phase. For final testing, the quad initially used for development was added to the training set, which yielded 4 final folds for testing.

3) Results And Discussion: Table VIII shows the results for the development data and Table IX for the evaluation data. It is clear that only spectral flux is the only feature which gives anything above chance level. It is hard to speculate on the reason for this, but it should be pointed out that data sparseness, i.e. very few competitive samples, may have contributed to this.

G. Classifier IIb: Outcome of Overlap

The observed outcome from overlap is defined by a contextual timing feature. This feature is the end-time of the talk-spurt for the speaker who intercept in the overlap subtracted by the end-time of the talk-spurt of the interlocutor, which is the speaker who talked before the overlap. Thus, this feature measures the outcome of the overlap, i.e the winner of the floor, and is hence denoted as the outcome. This is illustrated in Figure 7. Based on the outcome, the following labeling scheme is applied:

```
If outcome < 0 then
label as incoming speaker stops;
else
label as incoming speaker continues.
```

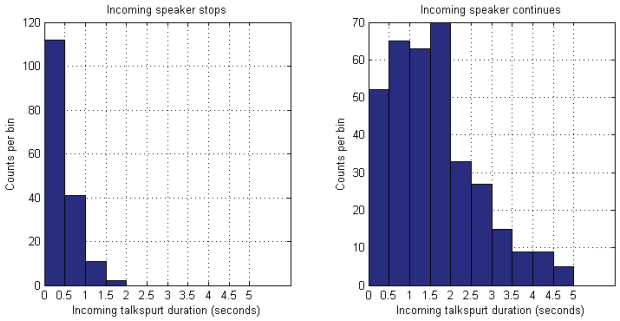


Fig. 8: Durations of talkspurts in overlap with no MTACK context (within the overlap). To the left is when the incoming speaker stops, and to the right is when the incoming speaker continues.

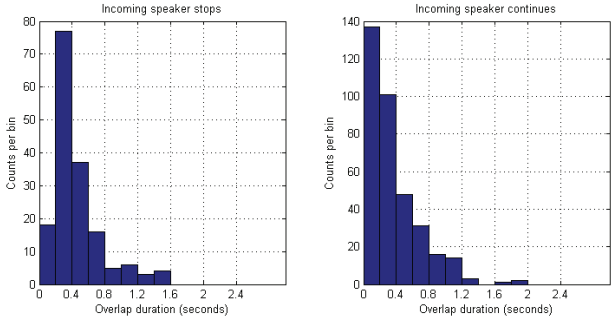


Fig. 9: Durations of overlaps with no MTACK context (within the overlap). To the left is when the incoming speaker stops, and to the right is when the incoming speaker continues.

By using this rule, instead of human annotations of interruptions, or competitive and cooperative speech, the resulting labels are always consistent and objective. If the labels generated by the rule may be predicted using acoustic cues, then the predicted outcome from the overlap can be forwarded to the dialog manager, which in turn can make a decision. In this way, we can think of the rule as an observed habit which may be predicted. However, the labels produced by the rule has no correspondence with the labels derived from annotation, the average F-score is 41.8.

Theoretically, one would expect a relation between the outcome of the overlap described here, on the one hand, and the concept of COMPETITIVE vs COOPERATIVE described earlier, on the other hand: the Speaker will probably more often stop speaking due to incoming COMPETITIVE Responses than due to COOPERATIVE Responses. Figures 8 and 9 show the histograms of the talk spurt durations and the overlap durations for the two possible outcomes of overlap. Compare these with Figures 4 and 5 to see that at least in this respect, there is a relation between observed outcome of overlap, and the manual annotation of COMPETITIVE vs. COOPERATIVE.

1) Acoustic Features: The final acoustic feature set is:

- F0: DCT 1-6
- Intensity: DCT 0-6 or 1-6
- Duration: For training, the full talk-spurt duration was used, for testing, the duration up to the maximum latency threshold was used.
- Spectral flux: 0'th DCT

TABLE X: Development set Average F-scores for predicting outcome of overlap given the "ideal VA talk spurts"

Max lat.(ms)	300	500	700	900	1100
F0	56	66	65	67	69
Int.	58	63	61	59	63
Int. + 0th	63	63	61	64	66
sp. Flux	61	62	62	62	64
v.q.	58	62	65	65	64
dur.	70	75	76	76	76
comb1	57	62	66	66	66
comb2	54	66	71	73	74
comb1 rs	58	63	63	67	65
comb2 rs	60	63	69	77	71

TABLE XI: Evaluation set Average F-scores for predicting outcome of overlap given the "ideal VA talk spurts"

Max lat.(ms)	500	700	900	1100
dur.	77	79	79	79
comb1	52	54	55	58
comb2	62	67	70	63
comb1 rs	53	60	56	61
comb2 rs	58	71	77	74

- Voice Quality: 0'th DCTs of spectral centroid, spectral kurtosis and spectral skewness.
- Comb1: F0,Intensity, Spectral flux and Voice Quality, as specified above
- Comb2: As Comb 1 with duration added, as specified above
- Comb1: Comb 1 with rescaling
- Comb2: Comb 2 with rescaling

The DCT coefficients are computed as described in Section VII-B.

2) Results And Discussion: The results, measured by average F-scores, for optimal parameters on the development set given the "ideal VA talk spurts" are shown in Table X. It is clear that performance increases with the maximum latency duration threshold. Adding the 0'th DCT coefficient to Intensity gives some benefit, but it is not included in the combined feature set since it might be sensitive to recording conditions. Duration is the most salient feature overall while the other features gives similar contributions. Rescaling does not show any clear advantage. Eventually, we decided to evaluate the combined feature sets, with and without rescaling, and, finally, duration alone.

The results for the evaluation set are given in Table XI. These results verify that classifier performance increases with the maximum latency duration threshold. Rescaling gives a clear advantage, but the comb2 feature set does not beat duration alone. Especially, the results the comb1 feature set (acoustic features only), are not very strong but clearly above chance for longer maximum latency thresholds.

Then we made the evaluation using the "OpenSmile VA talk spurts", the performance dropped significantly. The cause was hypothesized to be inconsistent segmentation by the energy based voice activity detector. Since he trajectory parametrization by DCT coefficients is likely to be sensitive to segmentation inconsistencies, we decided to only use the 0th DCT coefficient (i.e. corresponds to the arithmetic average). However, this ruled out using F0 and Intensity as features since the arithmetic average of these are dependent on the speaker and the distance between the speaker and the microphone. Consequently, we ended up using the 0th coefficients of Spectral Flux and Voice Quality. The results are shown in Table XII. It is clear that the acoustic features does not perform above chance, leaving only duration as a reliable feature.

TABLE XII: Evaluation set Average F-scores for predicting outcome of overlap given the "OpenSmile talk spurts"

Max lat.(ms)	500	700	900
dur.	66	71	69
comb1	N/A	48	46
comb2	54	54	49

H. Classifier IIc: Hybrid approach

The pragmatic approach in Section VII-G doesn't produce automatic labels that relate to the labels from the annotation. This section describes an attempt to derive a low complexity rule which shows agreement with the labels derived from the human annotations.

Similar to the pragmatic approach in Section VII-G, two types of contextual timing features are defined first. The first one is the duration of the overlap. The second is the end-time of the talk-spurt for the speaker who intercept in the overlap subtracted by the end-time of the talk-spurt of the interlocutor, which is the speaker who talked before the overlap. Thus, this feature measures the outcome of the overlap, i.e the winner of the floor, and is hence denoted as outcome.

To derive a rule from the features a decision tree was used, where the priors for the agreed labels were set to a uniform distribution. The first two rules, at the top of the tree, was:

```
If overlap > 0.15 and outcome < -0.40 then
label as competitive;
else
label as cooperative.
```

This label scheme achieved an average F-score with our agreed labels of 0.67. The value is above chance and should be compared to the kappa which is decent but not high. Rules with higher complexity may be derived by looking further down into the tree, but these high complexity rules are difficult to explain and understand.

The part of the rule which concerns the amount of overlap, i.e. a minimum overlap of 150 ms, may be interpreted as the minimum duration of a perceivable overlap. Thus, if the overlap is below 150ms it is not perceivable and hence no interruption is perceived either. It should be noted, that despite no listener responses, as defined by the acknowledgments moves, were included for annotation, more than a few listener responses where observed during annotation. These also tend to come immediately before the end of the talk-spurt, possibly within the last 150 ms. Since listener responses are considered as cooperative, the occurrences of these just toward the end of a talk-spurt may be another explanation for this criterion. In any case, a speaker change that is within 150ms before the end of the talk-spurt may simply be considered as a smooth speaker shift. Also, this criterion seem to be non-negligible, since if this part of the rule was removed, the average F-score dropped below chance level. The second part of the rule; outcome < -0.40; simply states that the speaker who intercepts in overlap has to speak for 400 ms after the overlap in order to consider it to be competitive. Finally, we notice that the rule implies a minimum talk-spurt duration of $150 + 400 = 550$ ms. We further notice from Figure 2, that listener responses are more likely to be shorter than 500ms compared to non-listener responses. This confirms the findings by [30], where duration was found to be a highly reliable feature for back-channels.

Figures 10 and 11 show the histograms of the talk spurt durations and the overlap durations for the labels generated by the rule. We notice a greater similarity between these histograms and the histograms for the manual annotations (Figures 4 and 5) compared

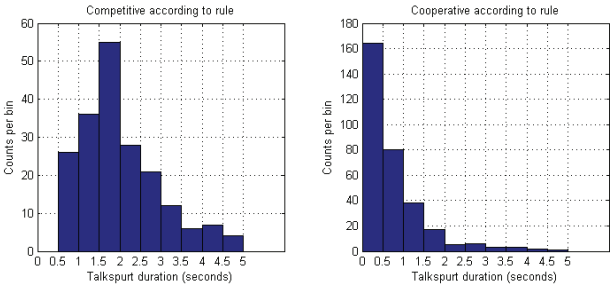


Fig. 10: Durations of talkspurts in overlap with no MTACK context (within the overlap). To the left are COMPETITIVE and to the right COOPERATIVE Responses, both according to the rule.

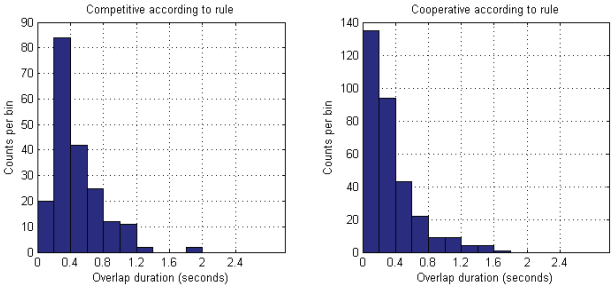


Fig. 11: Durations overlaps with no MTACK context (within the overlap). To the left are COMPETITIVE and to the right COOPERATIVE Responses, both according to the rule.

to the histograms which results from the pragmatic approach. This is especially true for the overlap durations of competitive speech.

To summarize, the motivations for using the hybrid rule are:

1) The rule extracts labels which have some consistency with our human annotations.
2) The rule generates labels which have overlap and duration distributions similar to the human annotations.
3) We can generate labels for more data than what is provided by the annotations.
4) The rule is always consistent and objective.

If the labels generated by the rule may be predicted using acoustic cues, then the predicted labels can be forwarded to the dialog manager, which in turn can make a decision. In this way, we can think of the rule as an observed habit which is also related to cooperative and competitive speech, which may be predicted.

1) Results And Discussion: For this classifier, we use exactly the same feature set as for the pragmatic approach (Section VII-G).

TABLE XIII: Development set Average F-scores for predictingCOMPETITIVE speech based on the hybrid approach given the "ideal VA talk spurts"

Max lat.(ms)	300	500	700	900	1100
F0	57	64	64	66	67
Int.	61	67	63	64	67
Int. + 0th	61	64	63	69	72
sp.flux	63	66	64	62	62
v.q.	62	64	67	68	69
dur.	41	71	79	79	82
comb1	61	67	66	64	67
comb2	50	58	62	65	67
comb1 rs	60	64	66	72	70
comb2 rs	55	58	69	75	76

TABLE XIV: Evaluation set Average F-scores for predicting COMPETITIVE speech based on the hybrid approach given the "ideal VA talk spurts"

Max lat.(ms)	500	700	900	1100
dur.	67	74	70	81
comb1	57	57	60	60
comb2	55	61	55	62
comb1 rs	58	57	58	62
comb2 rs	56	63	61	67

TABLE XV: Evaluation set Average F-scores for predicting COMPETITIVE speech based on the hybrid approach given the "OpenSmile talk spurts"

Max lat.(ms)	500	700	900
dur.	57	63	67
comb1	52	53	49
comb2	48	42	47

The results, measured by Average F-scores, for optimal parameters on the development given the "ideal VA talk spurts" are shown in Table XIII. The F-scores pretty much follows the same pattern as for the pragmatic approach (Section VII-G), but the observations are rephrased where for clarity with few but some differences. It is clear that classifier performance increases with the maximum latency duration threshold. Adding the 0'th DCT coefficient to Intensity gives some benefit, but it is not included in the combined feature set since it might be sensitive to recording conditions. Duration is the most salient feature overall while the other features gives similar contributions. Rescaling does show an advantage for maximum latency threshold of 700 ms and above. Eventually, we decided to evaluate the combined feature sets, with and without rescaling and finally duration alone.

The results for the evaluation set are given in Table XIV. These results verify that classifier performance increases with the maximum latency duration threshold. Rescaling gives a clear advantage, but the comb2 feature set does not beat duration alone. Especially, the results the comb1 feature set (acoustic features only), are not very strong but clearly above chance for longer maximum latency thresholds.

For the evaluation using the "OpenSmile VA talk spurts", we adopted the same procedure as for the pragmatic approach. Thus, we ended up using the 0th coefficients of Spectral Flux and Voice Quality along with duration. The results are shown in Table XII. It is clear that the acoustic features does not perform much above chance, leaving only duration as a reliable feature.

I. Conclusions from Classification Experiments

These series of experiments has shown successes and failures. First of all, **Classifier I** (Classification of all Responses into MTACK / NONMTACK) has a clear potential in a fielded system. For the **Classifier II b/c** versions, we have shown some success for acoustic features by using "ideal VA talk spurts". However, under the more realistic condition where "OpenSmile talk spurts" are used, only duration showed to be a reliable feature. It is not obvious to chose between **Classifier IIb** and **Classifier IIc**, mainly because the actual performance is similar, but the more more pragmatic **Classifier IIb** may be the choice since it does not rely on human judgments. Finally, it should be noted that all these classifiers may run in parallel for different maximum latency thresholds. Then different decision thresholds may be applied for the more reliable classifiers, which usually are the ones which has a higher maximum latency.

VIII. Existing Models for Behavior Generation and Specification

Here we describe Elckerlyc, the BML Realizer used to generate virtual human behavior. It is based on the SAIBA Framework [31] (see Fig 12, which describes a generic architecture for virtual human applications. It contains a three-stage process: *communicative intent planning*, multimodal *behavior planning*, resulting in a BML stream, and *behavior realization* of this stream. The Elckerlyc framework used in this project encompasses the realization stage. It takes a specification of the intended behavior of a virtual human written in the Behavior Markup Language (BML) [31] and executes this behavior through the virtual human.

Fig. 12: The SAIBA framework.

The BML stream contains BML requests with behaviors (such as speech, gesture, head movement etc.) and specifies how these behaviors are synchronized (see also Fig. 13). Synchronization of the behaviors to each other is done through BML constraints that link synchronization points in one behavior (start, end, stroke, etc; see also Fig. 14) to synchronization points in another behavior. BML can be used to append or merge new behaviors into a running BML stream. Some extension have been proposed to allow the specification of instant removal of a running BML request[1].

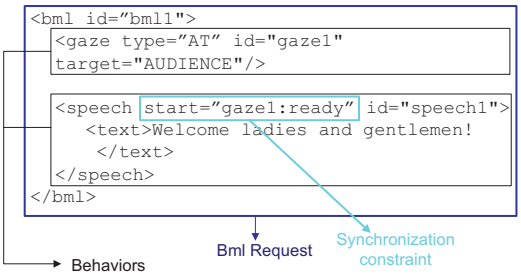


Fig. 13: An example of a BML request containing a gaze and a speech behavior. A synchronization constraint ensures that the speech starts after the gaze is aimed at the audience.

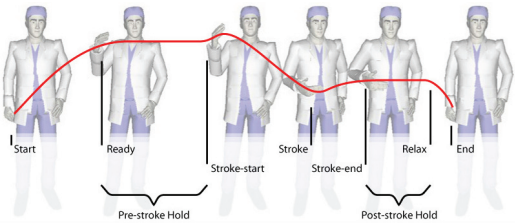


Fig. 14: Standard BML synchronization points (picture from http://wiki.mindmakers.org/projects:bml:main)

IX. Scheduling and Planning for Continuous Interaction

Currently, BML does not contain mechanisms to slightly modify behavior that is already running, or to interrupt behavior in a more graceful manner. Such mechanisms are crucial to achieve continuous interaction [32]. Some desired changes to planned behavior are only on their timing or parameter values (speak louder, increase gesture amplitude) and should not lead to completely rebuilding the animation or speech plan. Such small adaptations of the timing or shape of planned behavior occur in conversations and other interactions [2]. Elsewhere, we discuss the specification and Elckerlyc's implementation mechanisms that allow such small behavior plan changes to occur instantly [32]. In this paper we focus on graceful interruption and preplanning of behavior that were developed during the eNTERFACE workshop.

We have defined a custom BML extension BMLT [2] to allow the expression of behaviors and the scheduling and interruption mechanisms discussed above that cannot be expressed in BML (yet).

A. Preplanning

Planning a BML request typically takes a non-neglectable amount of time, especially if the timing of speech is to be obtained through speech synthesis software. This is problematic for developing highly responsive Virtual Humans like the one described in this paper. Elckerlyc explicitly models the scheduling stage of BML requests and makes it transparent to the Behavior Planner by providing it with feedback on when the scheduling of a BML request is started and when it it done. BMLT provides *preplanning* as a mechanism to construct a behavior plan that can be activated later on. In a typical usage scenario of pre-planning, the Behavior Planner already knows what behavior to execute, and wants to execute it (near) instantly later on, for example in reaction to some event such as an incoming Response from the user. Preplanning is set up for a BML request, using the BMLT preplan attribute in that request. Preplanned BML requests can be activated using another BML request with an onStart attribute. The preplanned behavior is activated as soon as the scheduler finishes planning the behavior with the `onStart` that activates it. Example 1 illustrates the BML used for preplanning.

BML Example 1 Several BML requests illustrating the preplanning and activation of pre-planned behavior.

```
<bml xmlns:bmlt="http://hmi.ewi.utwente.nl/bmlt"
  id="bml1" scheduling="merge" bmlt:preplan="true">
...
</bml>
```

(a) Preplan bml1.

```
<bml xmlns:bmlt="http://hmi.ewi.utwente.nl/bmlt"
  id="bmlX"
  bmlt:onStart="bml1"/>
```

(b) Activate preplanned behavior bml1.

```
<bml id="bml3"
xmlns:bmlt="http://hmi.ewi.utwente.nl/bmlt"
  scheduling="append-after(bml2)"
  bmlt:onStart="bml1,bml5">
...
</bml>
```

(c) Schedule `bml3` to be appended after `bml2`, activate preplanned behaviors `bml1` and `bml5` as `bml3` is started.

B. Graceful interruption

The interrupt behavior, first proposed and implemented in the SmartBody BML realizer [33], is used to interrupt a running BML request. This can be used to schedule the interrupt of a BML request relative to some other behavior (e.g. VH looks at the interlocutor

[1]See http://wiki.mindmakers.org/projects:bml:multipleblockissue

[2]See http://wiki.mindmakers.org/projects:bml:bmlt

before it stops to speak). In both BMLT and the SmartBody BML, the interrupt behavior by default immediately interrupts all behaviors in the BML request it targets at the start of the interrupt behavior.

In its simplest form (See Example 2), the BMLT interrupt behavior acts the same as the SmartBody interrupt behavior. The syntax is also very similar.

BML Example 2 Interrupt `bml1` as soon as `shake1:stroke` is reached

```
<bmlt:interrupt id="interrupt1"
target="bml1" start="shake1:stroke"/>
```

We have extended the interrupt behavior to allow a more fine-grained interrupt specification, using the `interruptspec` element inside an `interrupt` behavior. Using the `interruptspec` we can define exactly when certain behaviors inside the target BML request are to be interrupted. All behaviors in the target BML request that are not described in an `interruptspec` are interrupted instantly. The `interruptspec` also allows us to specify preplanned BML requests that are to be activated as soon as a certain behavior is interrupted using the `onStart` attribute. This combination of the interruption behavoir and preplanning allows us to specify the graceful interruption of behavior in other BML blocks, with alternative continuations after the interruption (See Example 3).

BML Example 3 The realizer interrupts all behaviors in `bml1`. `speech1` is interrupted at `sync1` and gracefully ended with some trailing speech using `bml3`, `gesture1` is interrupted at its `stroke-end`, and followed by the content of `bml4`. All other behaviors in `bml1` are interrupted at the start of `interrupt1` (that is, at `shake1:stroke`).

```
<bmlt:interrupt id="interrupt1"
target="bml1" start="shake1:stroke">
  <bmlt:interruptspec behavior="speech1"
   interruptSync="sync1" onStart="bml3"/>
  <bmlt:interruptspec behavior="gesture1"
   interruptSync="stroke-end" onStart="bml4"/>
</bmlt:interrupt>
```

X. Listener Response Elicitation

Before going into monitoring and handling Responses it is important that the system is able to elicit these Responses. In human-human conversation the speaker often elicits such responses. The speaker creates Response opportunities through vocal and non-vocal cues, such as pausing between statements, modifying the prosody of the speech, and using gaze and face expressions. This section discusses the literature in order to find possibilities for response elicititation cues that can be used in our pilot experiment.

Prosodic elicitation cues for responses are quite well described in literature. Gravano and Hirschberg [34] observe that the final intonation of the interpausal unit (IPU) preceding a response rises in 81% of the cases. Furthermore the mean intensity and pitch level of the preceding IPUs which are followed by a response are higher than IPUs not followed by a response. Furthermore Ward and Tsukahara [35] use in their handcrafted rule based model an area of 110ms of low pitch to predict a response 700ms after this cue.

Nonverbal cues are far less concretely described in literature. Such work mostly concerns gaze behavior. In a detailed study Bavelas et al. [36] conclude that 83% of listener responses in their corpus occur during mutual gaze, confirming earlier intuitions of Kendon [37] and Duncan Jr. [38]. Furthermore, head movements have been associated with eliciting responses [39], but there are, to our knowledge, no concrete findings directly applicable to virtual humans.

We performed an observatory study on the MultiLis corpus, where we analyzed the speakers who elicited the most responses from the listeners, with special attention to their nonverbal behaviors. Some speakers were very expressive in their nonverbal behavior, while others were not. For one of the speakers his blinking behavior really stood out. In general his blinking rate was high, but at the end of statement, where he expected a response from the listener, he stopped blinking and stared at the listener. He started blinking again as soon as the listener provided a response.

A. Enhancing MARY TTS to realize vocal elicitation cues

The MARY TTS platform is an open-source, modular architecture for building text-to-speech systems, including unit selection and statistical parametric waveform synthesis technologies. It has been described in detail elsewhere [40], [41]. The present paper only describes the aspects relevant in the current context. One of those aspects is how to realize vocal elicitation cues using MARY TTS. Prosody modification techniques are the key to realize vocal elicitation cues. Traditionally in MARY, the applications that require control over prosody were using MBROLA diphone synthetic voices, though the voices are unnatural. Nowadays HMM-based voices are reaching high quality synthetic speech.

In HMM-based speech synthesis, trained statistical models (context-dependent HMMs) are used to predict duration and generate parameters like mel-cepstral coefficients, log F0 values, and bandpass voicing strengths using the maximum likelihood parameter generation algorithm including global variance [42]. In the later stages, F0 parameters, bandpass voicing strengths, and the five bandpass filters are used to generate a mixed excitation signal. Finally, speech is synthesized from the mel-cepstral coefficients and the mixed excitation signal using the MLSA filter [43].

Although MARY already supports realization of predicted prosody parameters using HMM synthesis, it did not support explicit prosody specification. This project requires support for prosody modifications specified in MARYXML requests. So, as part of this project, we implemented support for 'prosody' element as described in W3C Speech Synthesis Markup Language (SSML) recommendations; and the different attributes in 'prosody' element like 'rate', 'pitch' and 'contour' are used as specifications to modify predicted phone durations and pitch contour before passing them to the HMM synthesizer. Once the modifications are done according to given specifications, they are realized as normal with HMM-based synthesis strategies.

MARYXML Example 1 An example which supports explicit prosody specifications

```
<?xml version="1.0" encoding="UTF-8" ?>
<maryxml version="0.4"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://mary.dfki.de/2002/MaryXML"
 xml:lang="en-US">
 <p>
  <prosody rate="fast"
     pitch="+10%"
     contour="(10%,low)(80%,+10%)(100%,+5st)">
   Welcome to the world of speech synthesis!
  </prosody>
  </p>
</maryxml>
```

XI. Pilot Experiment

As a setting for our experiments we chose the route description domain. This domain was chosen since in this domain the fact whether the information given by the agent has reached the user,

and is understood by the user or not, is crucial to the success of the interaction. In this setting, continuous monitoring of the user and reacting appropiately to their responses is very relevant. You may want to repeat certain elements of the explanation to get your point across or skip a part depending on the actions of the user.

Before going into monitoring and handling the responses it is important that your system is able to elicit these responses. In human-human conversation the speaker often elicits such responses. The speaker creates response opportunities by providing eliciting cues to the listener, such as pausing between statements, modifying the prosody of the speech and displaying various nonverbal behaviors. In this experiment we aim to recreate these signals based on literature and corpus analysis and evaluate them in our agent to see which elicitation strategy elicits the most responses. Furthermore we assess each version of our agent on subjective measures related to conversational skill, rapport, personality etc.

A. Task

During the experiment our route giving agent explains a route to the participants. Afterwards the participant needs to draw the route on a map, which is presented before the interaction begins.

B. Stimuli

The map contains the layout of a fictional city. Landmarks are highlighted on the map, such as a cathedral, a stadium, and bridges. With the map comes a legend explaining the terminology used by the agent to identify the landmarks. The current position of the participant is also shown on the map.

There are three different starting points, for three different routes. Each route consists of n steps[3] that take the user to their final destination. Each step is realized by specifying a BML block. The BML block specifies the speech and the behavior the agent performs. The speech is synthesized using Mary TTS [41]. The speech is manually cleaned up, using the prosody tags described in Section X. We removed, where necessary, peculiarities in the synthesized speech, added some extra pause moments and changed the speech rate, to make the agent sound more natural. Aligned with the speech, gestures are added to accompany the explanation of the route (e.g. pointing to the left or making an iconic gesture representing a landmark). The pause between the blocks is 1.5s, which is based on the mean pause between statements in the MultiLis corpus.

These pauses between the blocks are the response opportunities where we explicitly elicit responses. For each route we created four versions, each with different response elicitation behavior. These four different behavior are:

- **Default**: No explicit elicitation behavior.
- **Vocal**: Rising pitch at the end of the step.
- **Nonverbal**: Emphasis head and face gestures, interruption of blinking and gaze away as conformation behavior.
- **Combined**: Combination of the Vocal and Nonverbal behavior.

In the *Default* version no explicit elicitation behavior is employed. This version was our baseline from which we created the three following versions, by changing the pitch contours, or adding extra behaviors according to strict rules.

In the *Vocal* version we modified the pitch of the speech. The modification were inspired by Gravano and Hirschberg [34]. In their analysis of the Columbia Games Corpus, which is a task-oriented corpus, comparable to our setup (as opposed to spontaneous dialogues), they concluded that, among other features, the rising of the pitch in the final 200 to 300ms of speech is a response eliciting cue. We applied this finding to our synthesized speech in this version, by giving the last word of a step in the route a rising pitch contour.

In the *Nonverbal* version we added nonverbal inviting behavior found in the MultiLis Corpus [13]. More specifically we choose one of the speakers and recreated his nonverbal response eliciting behavior. This speaker was chosen by looking at the top 5 speakers with the highest rate of elicited responses per minute and selecting the speaker where nonverbal cues were most prominently present (according to our perception). His eliciting behavior was the following. He emphasizes the last word in a sentence by accompanying it with a subtle head nod and short eyebrow raise. At the same time he stops blinking (he generally has a pretty high blinking rate, so this actually stands out) and stares at the listener. As soon as a response is given, he starts blinking again and averts his gaze to formulate his next sentence. This behavior is recreated in the nonverbal version.

In the *Combined* version we combine both the vocal and nonverbal behavior changes to the default version.

C. Methodology

We invited 9 participants (8 male, 1 female, aged between 25 and 54, all non-native English speakers) to interact with our route agent. Participants are told that the agent is able to perceive and react to short vocal and nonverbal responses (like nodding, saying *"Uh-huh"*, or *"Yes"*).

Before each interaction the user was presented the map with the starting point of the route. This map is taken away before the interaction starts. During the interaction the route agent gave a route description to the user. It was the task of the user to remember the route and reproduce it on the map afterwards.

Each participant interacted three times with the route agent. During each interaction the agent explained a different route. Each route description was given with a different elicitation strategy. Every participant interacted with the *Default* and *Combined* agent and either the *Vocal* or the *Nonverbal* agent. Permutations of routes and elicitation strategies were varied among participants.

D. Measures

Before the experiment the participants filled in a prequestionnaire measuring their age, gender, native language and highest level of education.

After each route they filled out a questionnaire about the interaction. The questionnaire measures the rapport between the agent and the participant, based on the questionnaire used in De Kok and Heylen [13]. Furthermore we measured the perceived impression of the agent by having the participants rate the agent on 26 bipolar semantic differential adjective scales taken from the study of Ter Maat et al. [44]. All questions are on a 7-point Likert scale.

In the postquestionnaire after the final route, we asked which version of the agent they liked best, they thought was the most natural, the most social and the most attentive.

Our final measures are on the video recordings of the interaction. In these video recordings we counted the number and the type (nonverbal, vocal or both) of the responses they provided to the agent.

E. Results and Discussion

We succesfully elicited responses from the subjects (see Table XVI). The amount of response given seems highly subject dependent (see Table XVI). Over half of the subjects gave a response on all response elicitation positions in the route explanation, even if no explicit elicitation strategy was used. Perhaps the pauses between segments in the route explanations provide a very strong feedback

[3]For Route 1 and 3, $n = 8$, for Route 2, $n = 7$.

subject	default	combined	vocal	nonverbal	average
1	1	1	1	-	1
2	0.6	0.9	-	1	0.8
3	1	0.8	-	1	0.9
4	1	1	0.8	-	0.9
5	1	1	1	-	1
6	0.3	-	-	1	0.6
7	0.6	0.2	-	0.3	0.3
8	1	1	0.3	-	0.8
9	0.3	0.5	0.3	-	0.4

TABLE XVI: Response ratio (Responses given/Response opportunities in the route-description) per subject per elicitation strategy. The value '-' means that the specific elicitation strategy was not presented to the subject or that the recording failed.

	Default	Combined	Vocal	Nonverbal
Like best:	5 (56%)	3 (33%)	0 (0%)	2 (50%)
In between:	2 (22%)	4 (44%)	1 (20%)	1 (25%)
Like least:	2 (22%)	2 (22%)	4 (80%)	1 (25%)
Most natural:	5 (56%)	2 (22%)	1 (20%)	1 (25%)
In between:	2 (22%)	3 (33%)	1 (20%)	3 (75%)
Least natural:	2 (22%)	4 (44%)	3 (60%)	0 (0%)
Most social:	5 (56%)	3 (33%)	1 (20%)	0 (0%)
In between:	2 (22%)	4 (44%)	1 (20%)	3 (75%)
Least social:	2 (22%)	2 (22%)	3 (60%)	1 (25%)
Most attentive:	5 (56%)	3 (33%)	0 (0%)	1 (25%)
In between:	2 (22%)	5 (56%)	1 (20%)	1 (25%)
Least attentive:	2 (22%)	1 (11%)	4 (80%)	2 (50%)

TABLE XVII: Results of the post-questionnaire in which the participants ranked the agents on likeability, naturalness, social ability and attentiveness. Especially the agent with the *Vocal* elicitation strategy performs bad on these scales. The *Default* agent seems best.

elicitation cue. Only 6 out of 237 responses were non-verbal only. 137 were both verbal and nonverbal.

We observed the use of one or more repetions in the responses of five of the subjects (cf. Interaction Example 2).

Interaction Example 2 Example of repetition in the recordings.
Virtual Human: Take the second street on your right.
Subject: second street on my right.

Non-understanding was expressed in both intrusive (13x, for example: "over the square with the what?") and non intrusive ways (5x, for example: hesitant feedback: "Oh.. Keeeey" or with a puzzled look).

If we look at the result of the post-questionnaire (presented in Table XVII we notice the bad performance of the agent with the *Vocal* elicitation strategy. Most of the five participant that interacted with this agent rated it the lowest on all scales. The prosodic modifications to the speech to elicit responses should thus be improved. Now they are perceived as very unnatural. These modification also have a negative influence on the *Combined* elicitation strategy, since in this condition the same prosodic modifications are used. We think this is the reason why *Default* is generally considered the best condition on these measures.

The questionnaire after each session did not yield any insightful results.

F. Lessons learned

From the results of the pilot we learned that several improvements can be made to the setup. First we want to expand the experiment with a fourth route. This was always our intention, in order to let every participant interact with every elicitation strategy, but due to time constraints we decided to drop one of the routes for the pilot.

Furthermore the vocal elicitation strategy needs some work. On the postquestionnaire it was consistently rated as the least likable, natural, social and attentive of the four strategies. Since the vocal elicitation strategy is also included in the combined strategy, it probably had a negative impact on that condition as well.

Finally, we want to vary the pause between two sentences, since pause in itself is also a response elicitation cue [45], [46]. At this moment this pause is 1.5 seconds, based on the average pause in the MultiLis Corpus. We see in our data that in almost every response opportunity we explicitly created, we get a response. We suspect that the length of the pause is such a strong cue that this dominates our four different strategies and is the cause for this.

XII. Discussion and Conclusions

In this Enterface workshop, we have developed a virtual human that is able to interact with a 'real' subject in an continuous manner. That is: being capable of interaction in which all partners perceive each other, express themselves, and coordinate their behavior to each other, continually and in parallel. The project resulted in progress on several aspects of continuous interaction such as flexible and adaptive scheduling and planning of multimodal behavior (speech, gestures, facial expressions) including graceful interuption, automatic real-time classification of listener responses and models for appropriate reactions to listener responses. We have set up a pilot experiment in which a virtual human interacts with a subject. The aim of the experiment was to elicit Response behavior, to provide us with more information on what user responses occur, and to serve as inspiration for further interaction models.

In this experiment, we have observed that some Responses given by our subjects are much shorter than the waiting time between steps; other Responses are much longer. Furthermore, Responses are not given at every Response Oppurtunity. Starting to speak through a repetition or waiting for a Response that is already finished confused some of our subjects. In a responsive version of the virtual human, we should add dynamic pauses: if no Response comes, continue speaking after a smaller wait. If feedback comes, the virtual human can wait until Response is finished. If a Response is cooperative it often makes sense to immediately continue speaking in overlap.

We have observed several repetitions from the listener, related to speech from the speaker. Detecting such repetitions is still an open issue. Since the repetitions often repeat the landmarks used in the route, perhaps the occurence of landmarks (as detected by a keyword spotter) could be used as one of the cues for the identification of repetions. Assumed that we can automaticly asses whether a response is a repetition, the preplanning mechanisms we have developed during the workshop can be used to generate an acknowledgment of the repetition (see Interaction Example 3).

Interaction Example 3 Handling repetition.
Virtual Human: Turn right before the obelisk.
Subject: right before the obelisk.
Virtual Human: Yes. Then turn left and cross the bridge.

A generic set of such acknowledgements (e.g., "that's correct", "yes", "uhhuh") can be preplanned and activated instantly when needed. If the route description after the acknowledgements is already

planned, Elckerlycs retiming mechanisms (see [47]) can be used to shift it in time so that a full replan of the route description is avoided.

Interruptions are detected as Competitive Responses by our classifier. If the subjects interrupts the Virtual Human (as in Interaction Example 4), his ongoing route description can be gracefully interrupted using mechanisms discussed in Section IX-B. We can either preplan all alternative explanations, or use in-between generic preplanned sentences to cover up the scheduling, like "Ok, let me explain that again".

Interaction Example 4 Graceful interruption.

Virtual Human: Turn left at the square with the obelisk. Then take the second ...
Subject: over the square with the what?
Virtual Human: [gracefully interrupts ongoing behavior, selects an alternative for "Turn left at the square with the obelisk"] So you enter the square, there is an obelisk at the center of the square.

In the current implementation we have not yet explored different strategies to handle Responses from the user. Depending on the type of behavior that we would like to realize such strategies are selected in concordance with a politeness strategy and certain personality traits (e.g., dominance or impatience). For example: a rude or dominant virtual human could explicitly ignore interruptive responses by speaking louder and leaning forward to keep the turn, while a insecure virtual human could explicitly wait for feedback after each of its utterances. Some of this strategies can potentially already be realized with the existing system (e.g. merge a lean forward behavior, wait for feedback then continue). Elckerlyc can modify parameter values of ongoing behavior in an adhoc manner, allowing changes to for example gesture amplitude or speech volume. We are currently exploring how such parameter value changes can be *specified* in a formal manner, either through BML or through another channel that communicates with Elckerlyc (See [32] for a more elaborate discussion on this topic).

XIII. Deliverables

The project has resulted in several software components, corpora and annotations, that will be made available to the public:

1) Automatic, real-time classifiers for Responses, implemented as openSMILE components [4]
2) The addition of cooperative/competative annotations in the MapTask corpus [12]
3) A motion capture corpus containing over 100 gestures related to route-giving [5]
4) Extensions that allow prosody modification in HMM voices in the open source speech synthesis system Mary, these will be included in its new release [6]
5) Several extensions and tools for the open source virtual human platform Elckerlyc [7], which will be included in its next release:
   a) A generic WoZ interface framework that allows the set up of Wizard of Oz experiments with Elckerlyc in an easy and flexible manner
   b) Implementation of preplanning and scheduling algorithms that allow gracious interruption of ongoing behavior
   c) Integration of Elckerlyc with the open source SEMAINE api [11], an open source middleware framework that allows easy connection of different modules in emotion-oriented systems.
6) An annotated video corpus of user-interactions with our virtual human during the pilot experiment

[4]http://sourceforge.net/projects/opensmile/
[5]Freely available at http://hmi.ewi.utwente.nl/mocapdb
[6]Available at http://mary.dfki.de/
[7]http://hmi.ewi.utwente.nl/showcases/Elckerlyc

Acknowledgment

The authors would like to thank the project advisors, Anton Nijholt, Dirk Heylen, and Stefan Kopp, for their support, Marc Schröder for enjoyable discussions and useful tutorials, Albert Ali Salah for organizing eNTERFACE'10, Ronald Poppe and Mark ter Maat for conceptual and practical support, and the GATE project and the NoE SSPNet for sponsoring this project,.

References

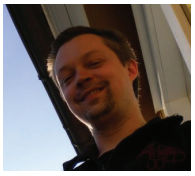
[1] K. R. Thórisson, *Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perception to Action*, ser. Multimodality in Language and Speech Systems. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2002, pp. 173–207.
[2] A. Nijholt, D. Reidsma, H. van Welbergen, H. op den Akker, and Z. M. Ruttkay, "Mutually coordinated anticipatory multimodal interaction," in *Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction*, ser. Lecture Notes in Computer Science, A. Esposito, N. G. Bourbakis, N. Avouris, and I. Hatzilygeroudis, Eds., vol. 5042. Berlin: Springer Verlag, 2008, pp. 70–89.
[3] D. T. Fujimoto, "Listener responses in interaction: A case for abandoning the term, backchannel," *Journal of Osaka Jogakuin 2year College*, vol. 37, pp. 35–54, 2007.
[4] D. Neiberg and J. Gustafson, "The prosody of Swedish conversational grunts," in *Proc. of Interspeech*, Sep. 2010.
[5] V. Manusov and A. R. Trees, ""are you kidding me?": The role of nonverbal cues in the verbal accounting process," *Journal of Communication*, vol. 52, no. 3, pp. 640–656, Sep. 2002.
[6] P. French and J. Local, "Turn-competitive incomings," *Journal of Pragmatics*, vol. 7, pp. 17–38, 1983.
[7] J. B. Bavelas, L. Coates, and T. Johnson, "Listeners as co-narrators," *Journal of Personality and Social Psychology*, vol. 79, no. 6, pp. 941–952, 2000.
[8] H. H. Clark and M. A. Krych, "Speaking while monitoring addressees for understanding," *Journal of Memory and Language*, vol. 50, no. 1, pp. 62–81, 2004.
[9] C. Goodwin, "Between and within: Alternative sequential treatments of continuers and assessments," *Human Studies*, vol. 9, no. 2-3, pp. 205–217, 1986.
[10] ——, *Conversational Organization: interaction between speakers and hearers*. Academic Press, 1981.
[11] M. Schröder, "The SEMAINE API: Towards a standards-based framework for building emotion-oriented systems," *Advances in Human-Computer Interaction*, vol. 2010, 2010.
[12] A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty-Sneddon, S. Garrod, S. Isard, J. C. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. Thompson, and R. Weinert, "The HCRC Map Task corpus," *Language and Speech*, vol. 34, pp. 351–366, 1991.
[13] I. de Kok and D. Heylen, "The MultiLis corpus – dealing with individual differences of nonverbal listening behavior," in *Proceedings of COST 2102: Toward Autonomous, Adaptive, and Context-Aware Multimodal Interfaces: Theoretical and Practical Issues*, 2010.
[14] P. T. Brady, "A statistical analysis of on-off patterns in 16 conversations," *The Bell System Technical Journal*, vol. 47, pp. 73–91, 1968.
[15] J. C. Carletta, S. Isard, G. Doherty-Sneddon, A. Isard, J. C. Kowtko, and A. H. Anderson, "The reliability of a dialogue structure coding scheme," *Computational Linguistics*, vol. 23, no. 1, pp. 13–31, 1997.
[16] D. Reidsma, "Annotations and subjective machines — of annotators, embodied agents, users, and other humans," Ph.D. dissertation, University of Twente, Oct. 2008.
[17] N. Ward, "Non-lexical conversational sounds in American English," *Pragmatics and Cognition*, vol. 14, no. 1, pp. 129–182, 2006.
[18] F. Eyben, M. Woellmer, and B. Schuller, "opensmile - the munich versatile and fast open-source audio feature extractor," in *Proceedings of ACM Multimedia*, 2010, to appear.
[19] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[20] J. Gustafson and D. Neiberg, "Prosodic cues to engagement in non-lexical response tokens in Swedish," in *DiSS-LPSS Joint Workshop 2010*, Sep. 2010.
[21] Y. Ariki, S. Mizuta, M. Nagata, and T. Sakai, "Spoken-word recognition using dynamic features analysed by two-dimensional cepstrum," *Communications, Speech and Vision*, vol. 136, no. 2, pp. 133–140, Apr. 1989.
[22] V. Tyagi, I. McCowan, H. Misra, and H. Bourlard, "Mel-cepstrum modulation spectrum (MCMS) features for robust ASR," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2003.
[23] D. Neiberg, P. Laukka, and G. Ananthakrishnan, "Classification of affective speech using normalized time-frequency cepstra," in *Prosody 2010*, May 2010.
[24] F. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the twenty-first international conference on Machine learning*, Banff, Canada, 2004.
[25] S. Benus, A. Gravano, and J. Hirschberg, "The prosody of backchannels in american english," in *Proceedings of the 16th International Congress of Phonetic Sciences 2007*, 2007, pp. 1065–1068.
[26] C. C. Lee, S. Lee, and S. S. Narayanan, "An analysis of multimodal cues of interruption in dyadic spoken interactions," in *Proceedings of Interspeech*, 2008, pp. 1678–1681.
[27] E. Schegloff, "Overlapping talk and the organization of turn-taking for conversation," *Language in Society*, vol. 29, pp. 1–63, 2000.
[28] E. Kurtic, G. J. Brown, and B. Wells, "Resources for turn competition in overlap in multi-party conversations: Speech rate, pausing and duration," in *Proceedings of Interspeech*, 2010, to appear.
[29] M. F. McKinneya, D. Moelants, M. E. P. Davies, and A. Klapuri, "Evaluation of audio beat tracking and music tempo extraction algorithms," *Journal of New Music Research*, vol. 36, no. 1, pp. 1 – 16, Mar. 2007.
[30] J. Edlund, M. Heldner, S. Al Moubayed, A. Gravano, and J. Hirschberg, "Very short utterances in conversation," in *Proceedings of Fonetik*, 2010.
[31] S. Kopp, B. Krenn, S. Marsella, A. N. Marshall, C. Pelachaud, H. Pirker, K. R. Thórisson, and H. H. Vilhjálmsson, "Towards a common framework for multimodal generation: The behavior markup language," in *Intelligent Virtual Agents, 6th International Conference*, ser. Lecture Notes in Computer Science, J. Gratch, M. R. Young, R. Aylett, D. Ballin, and P. Olivier, Eds., vol. 4133. Springer, 2006, pp. 205–217.
[32] H. van Welbergen, D. Reidsma, and J. Zwiers, "A demonstration of continuous interaction with Elckerlyc," in *Multimodal Output Generation*, 2010.
[33] M. Thiebaux, A. N. Marshall, S. Marsella, and M. Kallmann, "Smartbody: Behavior realization for embodied conversational agents," in *Proc. 7th International Conference on Autonomous Agents and Multiagent Systems*, 2008, pp. 151–158.
[34] A. Gravano and J. Hirschberg, "Backchannel-inviting cues in task-oriented dialogue," in *Proceedings of Interspeech*, Brighton, 2009, pp. 1019–1022.
[35] N. Ward and W. Tsukahara, "Prosodic features which cue back-channel responses in English and Japanese," *Journal of Pragmatics*, vol. 32, no. 8, pp. 1177–1207, 2000.
[36] J. B. Bavelas, L. Coates, and T. Johnson, "Listener responses as a collaborative process: The role of gaze," *Journal of Communication*, vol. 52, no. 3, pp. 566–580, 2002.
[37] A. Kendon, "Some functions of gaze direction in social interaction," *Acta Psychologica*, vol. 26, pp. 22–63, 1967.
[38] S. Duncan, Jr., "On the structure of speaker-auditor interaction during speaking turns," *Language in society*, vol. 3, no. 2, pp. 161–180, Dec. 1974.
[39] D. Heylen, "Head gestures, gaze and the principles of conversational structure," *International Journal of Humanoid Robotics*, vol. 3, no. 3, pp. 241–267, 2006.
[40] M. Schröder, M. Charfuelan, S. Pammi, and O. Türk, "The MARY TTS entry in the Blizzard Challenge 2008," in *Proc. of the Blizzard Challenge*, 2008.
[41] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.
[42] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 5, pp. 816–824, May 2007.
[43] A. W. Black, K. Tokuda, and H. Zen, "An HMM-based speech synthesis system applied to English," in *Proc. of 2002 IEEE SSW*, Santa Monica, CA, USA, Sep. 2002.
[44] M. ter Maat, K. P. Truong, and D. Heylen, "How turn-taking strategies influence users impressions of an agent," in *Proceedings of Intelligent Virtual Agents*, Philadelphia, Pennsylvania, USA, Sep. 2010.
[45] N. Cathcart, J. Carletta, and E. Klein, "A shallow model of backchannel continuers in spoken dialogue," in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 51–58.
[46] L.-P. Morency, I. de Kok, and J. Gratch, "A probabilistic multimodal approach for predicting listener backchannels," *Autonomous Agents and Multi-Agent Systems*, vol. 20, no. 1, pp. 70–84, May 2010.
[47] H. van Welbergen, D. Reidsma, Z. M. Ruttkay, and J. Zwiers, "Elckerlyc: A BML realizer for continuous, multimodal interaction with a virtual human," *Journal on Multimodal User Interfaces*, 2010, To appear.

Dennis Reidsma Dennis Reidsma is a postdoctoral researcher at the Human Media Interaction group. For his PhD he worked on different aspects of natural interaction systems. He worked, among other things, on problems of annotation and reliability in large multimodal annotated corpora, in the context of the EU FP6 AMI and AMIDA projects. In addition, he worked on research and development of new interactive systems with virtual humans. The *interactive virtual dancer* attempts to invite a human to engage with her, using computer vision, music analysis, and patterns of leading and following behavior. The *interactive virtual orchestra conductor* leads an ensemble of human musicians through a musical performance using advanced interactive graphics developed at HMI and advanced music processing algorithms. His current interests are in exploring continuous interaction with virtual humans in conversational settings. He is one of the Elckerlyc developers.

Khiet Truong Khiet Truong is a postdoctoral researcher at the University of Twente in the Human Media Interaction group. She has a background in computational linguistics and speech technology. As a Master student, she carried out research on automatic pronunciation error detection in speech of second-language learners. In 2009, she successfully defended her PhD thesis on automatic emotion recognition in speech based on work carried out at TNO. Currently, she is working on social signal processing in the SSPNet-project.

Herwin van Welbergen Herwin van Welbergen received his MSc in Human Media Interaction from the University of Twente's Department of Computer Science. Currently, he is a PhD candidate at the Human Media Interaction group. His research activities focus on real-time multimodal behavior generation for virtual humans, using real-time procedural animation, real-time physical simulation and speech, especially for applications that allow continuous interaction with a virtual human. Herwin is the main developer of the Elckerlyc framework.



Daniel Neiberg Daniel Neiberg received a Master of Science degree in electrical engineering in 2003 from KTH (Royal Institute of Technology), Sweden. He is currently a Ph.D. student at the department TMH at KTH. His fields of interest covers automatic affective recognition, conversational interaction, prosody and acoustic-to-articulatory inversion.

Sathish Chandra Pammi Sathish Pammi is a Researcher and PhD student in the DFKI LT lab. He has obtained his Masters degree in Computer Science Engineering from International Institute of Information Technology (IIIT, Hyderabad, India). He has joined the DFKI Speech Group in 2007. Since 2008, he has been working in the development of Text-To-Speech (TTS) systems, including synthesis of vocal listener behavior, for Sensitive Artificial Listeners (SAL) in EU FP7 SEMAINE project. He is one of the core developers of the MARY TTS system. His current research interests are interactive speech synthesis, conversational agents and talking robots.

Iwan de Kok Iwan de Kok studied Computer Science at the University of Twente, receiving his MSc. in 2009 for his thesis on the influence of videoconferencing and an emotional feedback support system on polyadic negotations. During his studies he did a 3 month internship at the USC Institute for Creative Technologies, working on the multimodal prediction of listener responses. Currently he is a PhD student in the Human Media Interaction Group of the University of Twente. His goal is to create a listener response generation model for virtual humans, with a focus on the nonverbal aspect.

Bart van Straalen Bart van Straalen is currently pursuing a PhD at the Human Media Interaction group at the University of Twente. The main focus of his PhD research is on the role of social and emotional capabilities on the selection and generation of communicative behavior in ECAs. As part of his research he works on contextual behavior analysis, modeling of cognitive processes and dialogue systems. His research interests lie in the area of cognitive modeling, emotion appraisal, coping strategies, FML-realization, artificial intelligence and human to ECA communication.

Vision Based Hand Puppet

Cem Keskin, İsmail Arı, Tolga Eren, Furkan Kıraç, Lukas Rybok, Hazım Ekenel, Rainer Stiefelhagen, Lale Akarun

Abstract— **The aim of this project is to develop a multimodal interface for a digital puppetry application, which is suitable for creative collaboration of multiple performers. This is achieved by manipulating the low- and high level aspects of 3D hierarchical digital models in real–time. In particular, the hands and the face of multiple performers are tracked in order to recognize their gestures and facial expressions, which are then mapped to kinematic parameters of digital puppets. The visualization of the puppet is provided as a feedback for the performer, and as an entertainment medium for the audience. Possible uses of this system include digital theaters, simplified animation tools, remote full-body interaction and sign-language visualization.**

The application consists of two separate hand tracking modules aimed at different shape and motion parameters, a facial feature tracker, a hand gesture and facial expression classification module, an XML based low-bandwidth communication module and a visualization module capable of handling inverse kinematics and skeletal animation. The methods employed do not depend on special hardware and do not require high computational power, as each module runs on separate computers.

I. Introduction

Puppetry is an ancient form of art and performance, which is known by most cultures in slightly different forms. Puppeteers either use sticks and ropes, or their bodies, as in hand puppetry, to animate the puppets. The forms of puppetry that do not require special devices are quite intuitive, and allows even a first time performer to succeed in animating a puppet in a visually pleasing manner.

Creative collaboration is common among most art forms, and puppetry can also be performed by multiple performers. Generally, multiple puppeteers manipulate separate puppets in order to form a theater play, but for some advanced puppets, several performers may be needed for a single puppet.

In digital puppetry, traditional puppets are replaced by 2D or 3D models that usually consist of several limbs, which can be manipulated separately and concurrently. In this work, we are concerned with 3D models that have a hierarchical skeleton with a high degree of freedom. Unless some high level animation parameters are defined, which act on several joints at the same time, these models are hard to manipulate using only low level parameters. This process is akin to digital animation, where animators create animations by carefully constructing the sequence frame by frame by manipulating each joint, which are then interpolated to form a fluent motion. This is a hard and time consuming process. Our aim is to create an intuitive interface, which allows non–expert performers to collaboratively manipulate a complex 3D model in real time.

Recent developments in technology allowed using motion capture devices to render moving humanoid models in a realistic manner. This technology is mainly used for commercial applications such as games and movies, and therefore, involves special worn devices and sensors. This method of capturing animation parameters is expensive and invasive. In this work, we are interested in estimating animation parameters using basic sensors without using markers and without the help of special devices. Previous work in this area includes CoPuppet, a system developed by Bottoni *et. al*, which makes use of gestures and voice and allows multiple users to act on a single puppet in a collaborative manner [Bottoni]. Whereas CoPuppet captures hand gestures using a special device, we use simple cameras and also allow facial expressions and head movements.

The main objective of this project is to design and implement a real-time vision-based digital puppetry system that does not rely on special sensors or markers. This work involves tracking of both hands, shape parameter estimation, motion tracking, gesture recognition, facial parameter tracking, expression classification, and also provides a graphical output that can give feedback about the efficiency and correctness of all the related modules in an eye pleasing and entertaining manner.

This report is structured as follows. In Section II we briefly describe the framework and its modules. In Section III, we give details of each module of the system. Particularly, in Section III-A we describe the stereo–vision based hand tracking module. In Section III-B, we provide the details of the facial expression detection and tracking module. Hand pose estimation module is described in Section III-C, and the recognition module is explained in Section III-D. The network protocol is given in Section III-E, and finally, the visualization module is explained in Section III-F. We provide discussions and mention our future work in Section IV.

II. System Description

This system is designed to allow multiple performers to collaborate in an intuitive manner. The only sensors used are cameras, and performers do not need to wear special markers. The number of puppeteers to perform is not limited, and they are not restricted to be at the same place. Each performer can communicate with the puppet over the network and get visual feedback at the same time.

Performers either use their hands or their faces to manipulate the puppet. In both cases, low level shape parameters are tracked, which are used to recognize certain hand gestures or facial expressions. Known gestures and expressions are used to give high level commands to the puppet, whereas shape parameters are directly used to manipulate certain limbs of the puppet.

Digital puppets are standardized by using a single skeleton for every model in order to allow seamless integration of new modules without complication. This minimizes the amount of knowledge that needs to be passed from the visualization module to the performer, as each puppet is virtually the same to the tracker module. Each module can manipulate each joint, and can give any high level command. Using a single skeleton does not constrain the shape of the puppet, but restricts the number of degrees of freedom that can be associated with a model. Specifically, we use a humanoid skeleton, which can be used to animate different objects, such as humans, animals, but also trees and buildings through rigging.

The most important criterion in choosing methodology is speed, as all the modules are required to run in real–time. The time it takes the puppeteer to perform and receive feedback should be minimal. Therefore, accuracy is sacrificed to allow rapid decision taking.

System flowchart is given in Figure 1. The framework uses three different tracking modules. The face feature tracking module uses a single camera facing the head of a single performer, and uses an active shape model to track certain landmarks on the face of the performer in real time. Hand pose estimation module uses multiple uncalibrated cameras to extract the silhouettes of the hand of the performer, and then tries to estimate the skeletal parameters that would conform to

C. Keskin, İ. Arı, F. Kıraç, and L. Akarun are with Boğaziçi University, Turkey. T. Eren is with Sabancı University, Turkey. L. Rybok, H. Ekenel and R. Stiefelhagen are with Karlsruhe Institute of Technology, Germany.

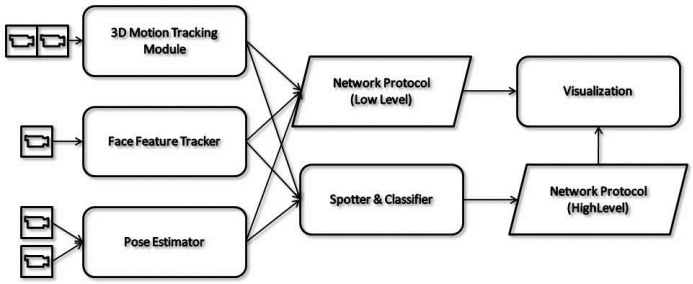


Fig. 1. System flowchart

all the silhouettes. 3D motion tracking module uses a stereo camera, or a pair of calibrated cameras to reconstruct the 3D trajectory of the hand. It also fits an ellipsoid on the estimated 3D point cloud for the hand, revealing more low level parameters associated with the hand shape.

Each of the tracking modules passes the parameter sequence to the recognition spotter and classifier module, which looks for known patterns in continuous streams of data. Face data is used to spot and recognize basic facial expressions, such as sadness and happiness. Motion data and the ellipsoid parameters retrieved from the 3D module is used to recognize 3D hand gestures. Likewise, the pose parameters supplied by the pose estimator module are used to spot certain hand posture–gesture combinations.

The visualization module continuously reads data coming from the modules and renders the puppet accordingly. The tracking and recognition modules send commands in the form of binary XML files over the network. The visualization module parses these and applies all the commands.

III. Methodology

A. Stereo–vision based hand tracking

In order to recognize hand gestures, the position of the hands first needs to be localized. To this end, we make use of a Bumblebee stereo camera system by Point Grey, allowing us the recovery of the 3D position of the hands. For hand localization, first skin-color segmentation is applied to the images captured with the left camera. Following the results form [1], we calculate for each pixel the probability of being skin-colored using Bayes' Theorem:

$$P(\text{Skin}|x) = \frac{P(x|\text{Skin}) \cdot P(\text{Skin})}{P(x)} \quad (1)$$

Here the class-conditional $P(x|\text{Skin})$ is modeled with a histogram trained on face images. Since in our scenario the hand is assumed to be the only visible skin-colored region in the image and is expected to occupy only a small fraction of it, the prior is set to $P(\text{Skin}) = 0.1$. An example of a skin-color probability map obtained using the described approach can be seen in Fig. 2.

For hand-detection, the skin-color map is thresholded, followed by the application of morphological operations to smooth out noise and skin-colored blobs are finally extracted using a connected component analysis algorithm. Further, the hand position is estimated by the center of the biggest blob. Finally, the area around the detected hand is matched in the right camera image using normalized cross-correlation and the so obtained disparity value is employed to calculate the 3D location of the hand.

Since tracking by detection is not stable enough and therefore results in noisy hand trajectories, the hand is tracked with a particle filter [2]. For the observation model again both skin-color and depth information are used. In order to achieve a low computational complexity the hand is modeled in 3D with a fixed-sized rectangle

Fig. 2. Example skin-color probability map used for hand tracking and detection

that is projected to the image plane for each particle (see Fig. 3) to evaluate the likelihood function.

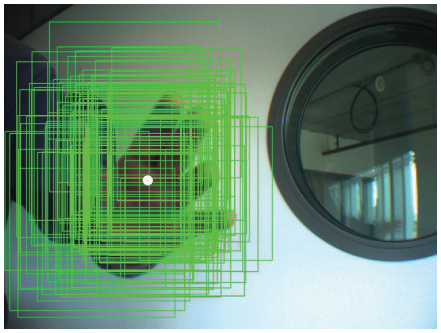


Fig. 3. Projected hand hypotheses (particles) of the tracker. The dot denotes the derived final hypothesis for the hand position.

The color cue is calculated by averaging the skin-color probability values in the region of interest defined by the projected particle multiplied by the number of pixels in the skin-color map that exceed a certain threshold. The multiplication ensures that particles that cover the hand region the most get a higher score than particles associated to a smaller region. The calculation of the depth score consists of a comparison of the disparity value defined by the particle and the disparity value obtained from matching the projected particle's area in the right camera image.

B. Vision based Emotion Recognition

The digital puppet is aimed to perform seven different emotion states in real time synchronously with the human performer. The chosen states are the six universal expressions (surprise, anger, happiness, sadness, fear, disgust) and the neutral facial expression.

With the promising results achieved in face and facial landmark detection research, emotion recognition started to take attention of the researchers especially in the last decade. Facial expression recognition and emotion recognition are used as overlapping terms by vision researchers since facial expressions are the visually apparent presences of internal emotions. Some surveys on the subject are available, such as Fasel and Luettin's review [3] on automatic facial expression analysis and Pantic and Rothkrantz's work [4] that examines the state of the art approaches in automatic analysis of facial expressions. Different approaches have been tried in facial expression analysis systems. All approaches share a common framework starting with face and facial landmark detection, facial feature extraction and expression classification. The main focus seems to be using static images whereas some papers discuss emotion recognition from image sequences, i.e. videos. For details, the reader may refer to Arı [5].

In this work, a similar way to Busso et al. is followed, where the authors report that they use commercial software for landmark tracking, partition the face into five regions of interest, create a histogram using PCA coefficients of these regions of interest and finally assign the video to one of the four expression classes using 3NN [6]. The whole system is run fully automatically and real time. First, we track the facial landmarks in face videos using Active Shape Model (ASM) based tracker which is modified from Wei's asmlibrary on Google code [7]. The landmark locations are used for the computation of high level features which is fed to the distance-based classifier which is an extended version of the nearest neighbor classifier.

1) Facial Landmark Tracking: ASMs are one of the state-of-the-art approaches for landmark tracking [8]. The ASM is trained using the annotated set of face images. Then, it starts the search for landmarks from the mean shape aligned to the position and size of the face located by a global face detector (in our case Viola-Jones face detector). Afterwards, the following two steps are repeated until convergence (i) adjust the locations of shape points by template matching of the image texture around each landmark and propose a new shape (ii) conform this new shape to a global shape model (based on PCA). The individual template matches are unreliable and the shape model improves the results of the weak template matchers by forming a stronger overall classifier. The entire search is repeated at each level in an image pyramid, from coarse to fine resolution using a multi-resolution approach. In the case of tracking, the model is initiated from the shape found on the previous frame instead of using the mean shape.

ASM performs better when person-specific model is trained. In this work, we had a generic model involving different subjects and a person-specific model which is trained from the face images of the test subject.

2) Feature Extraction: ASM-based tracker provides 116 facial landmarks which are seen on the left of Figure 4. Using the locations of the landmarks directly as feature seems not to be a good choice since the tracker works fine for many parts of the face such as eyebrows, eyes, chin, and nose, but not very robust for detecting the exact movements of the lips which is the most non-rigid part of the face. This phenomenon results from the fact that ASM models the face holistically and the small variations in the lips may be discarded during constraining by PCA. Moreover, the intensity difference on the lip boundaries are not as obvious as the other parts as seen in Figure 4. Thus, the landmark locations are used for computing 7 high level features seen on the right of Figure 4 as follows:

1) Average eye middle to eyebrow middle distance
2) Lip width
3) Lip height
4) Vertical edge activity over the forehead region
5) Horizontal edge activity over the lower forehead region
6) Sum of horizontal and vertical edge activity over the right cheek
7) Sum of horizontal and vertical edge activity over the left cheek

The first three features are computed using the Euclidean distance between the related landmarks. For the remaining features, the image is blurred with a Gaussian kernel and afterwards filtered with a Sobel kernel on horizontal and vertical axes separately. Then the average absolute pixel value is computed in each region. The vertical edge activity image is shown on the right of Figure 5 for surprise state. For example, the average pixel value residing in the forehead quadrilateral in the vertical edge activity image is found as the 4th feature.

In the setup we used, the test subject starts with neutral expression and waits in neutral state for about two seconds. The average values for the features are computed and the features in the remaining frames are normalized by multiplying by reciprocal of average.

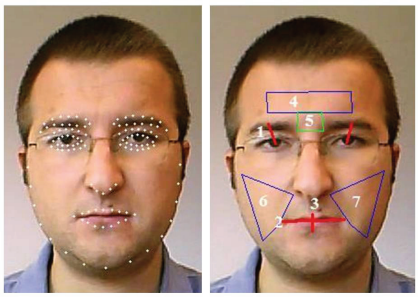


Fig. 4. Facial landmarks (on the left) and the regions of interest (on the right) used for feature extraction.

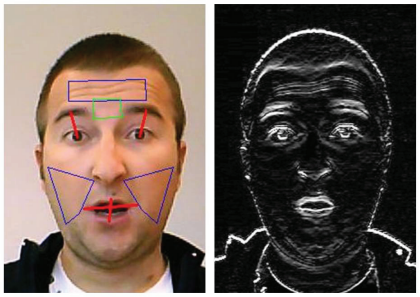


Fig. 5. A snapshot of surprise state (on the left) and corresponding vertical edge activity image (on the right).

3) Emotion Classification: Since the facial expressions vary with the subject and with the conditions of the environment such as illumination, we aimed a training setup which can be easily configured for a specific subject in a specific environment. During the training period, the subject starts by waiting in the neutral state and afterwards repeats each state five times. The interface records the feature vectors for these states, which are a total of 35 different vectors belonging to seven classes.

During testing, the subject again starts with neutral expression for normalization (and adaptation to environment). In the preceding frames, the feature vector is computed for each frame and then its average distance to the training feature vectors are computed for each class. Let $d_i, i = 1, , 7$ be the average distance computed for each class. The distances represent dissimilarity whereas $s_i = e^{-d_i}$ can be used as a similarity measure. Finally, s_i values are normalized such that their sum is one and they can be regarded as likelihood probabilities. This method is superior to nearest neighbor (NN) classification or kNN, since it performs soft assignment instead of hard assignment.

4) Extension for Unseen Subjects: For the training of ASM, 25 face images are gathered from the test subject. A generic model is used for automatic landmarking of the faces instead of annotating them from scratch. Then the locations of the landmarks are fine-tuned and the person specific model is trained from them. As mentioned in the previous subsection, the training of the classifier can be done easily for a specific subject under the current environmental conditions. This process can be followed for extending the system to work for new subjects.

5) Results: The results of the emotion recognition system can be seen in Figure 6. The likelihoods of the related emotional states are drawn on the user interface. A core2duo 1.2 GHz laptop computer with 3GB RAM can process about 15 frames per second with 320240 pixels resolution. The built-in laptop webcam is used.

The proposed system gives promising results for handling partial

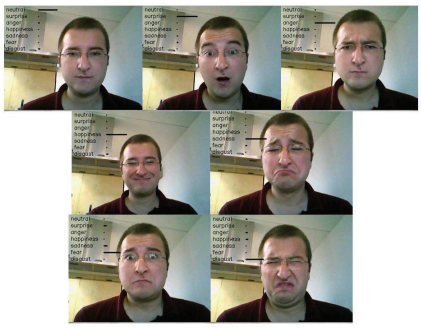


Fig. 6. Results of emotion recognition (neutral, surprise, anger, happiness, sadness, fear, disgust).

occlusions as seen in Figure 7.

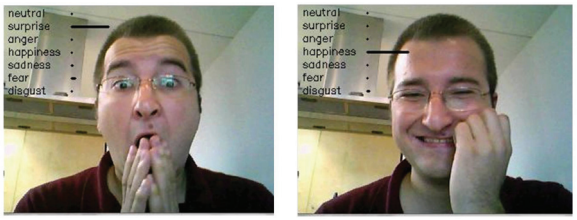


Fig. 7. Results of emotion recognition with partial occlusion.

The system is successful at discriminating seven different emotional states from a specific subject's video in real time. It is translation and scale invariant. The rotation invariance, on the other hand, depends on the performance of the tracker where ASM-based tracker provides the landmarks successfully for up to 20-30 degrees of rotation. The proposed system is open for improvements such as introducing more high level features on demand and extension for pose change.

C. Hand Pose Estimation Module

This module tracks the pose of a hand using silhouettes of the hands taken from two different cameras. Features are selected as silhouettes of a two camera setup. Dimensionality reduction is done by optimizing a Gaussian process latent variable model (GPLVM). For speeding up the optimization process Sparse GPLVM formulations have been used. Flowchart of the hand pose tracking module is shown in Figure 8.

1) Training Set Generation: The ground truth hand silhouette images for both cameras are generated by rendering a 3D hand model. "Poser" software's 3D hand object is manipulated through a Python script. The silhouettes are extracted and saved as PNG image files which then are loaded into Matlab for further training. An example of a hand silhouette taken from left and right cameras are shown in Figure 9.

2) Dimensionality Reduction Using GPLVM: The proposed hand tracking algorithm determines the hand pose from two silhouette images without ambiguity. Hand silhouette images are 80x80 pixel resolution images. Normally a feature extraction scheme would be applied to the silhouettes. However, in this case the pixels themselves are treated as features and given directly to GPLVM for dimensionality reduction. This provides an opportunity to test the quality of GPLVM's reduction. If GPLVM is able to capture the essence of the factors generating the silhouettes, then it will be able to capture a low dimensional manifold in the low dimensional latent space.

Considering each pixel as a unique feature of the hand, we have a 6400 dimensional feature vector per camera. Since two

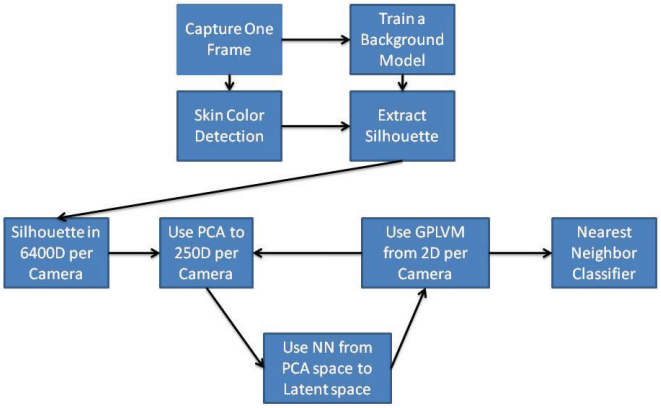


Fig. 8. Flowchart of the hand pose tracking module.

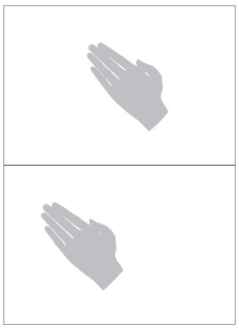


Fig. 9. An example of hand silhouettes as taken from left and right cameras respectively.

cameras are used the feature space is in 12800 dimensions. GPLVM requires an initialization step, a choice of a non-linear function for producing covariance matrices. Then a global search algorithm is applied to optimize the non-linear mapping. GPLVM is initialized with Probabilistic PCA (PPCA) and a radial basis function (RBF) kernel is used as the non-linear covariance function generator.

The captured manifold is represented in the latent space as below. Red crosses represent the silhouettes captured by the left camera and the green circles are the silhouettes captured from the right camera. 40 silhouette images per camera are used in the reduction phase.

As can be seen the silhouette manifolds for both of the cameras are extracted in a meaningful manner where there is only one ambiguity point in 2D latent space. This ambiguity should not be a problem. The tracking algorithm can be designed in a way to handle this kind of ambiguities. Since GPLVM finds a mapping from latent space X to the feature space Y but not the other way around, for tracking the hand pose, we have to generate new silhouettes from the generative model captured by the GPLVM and match the generated model with the captured silhouette. This action involves a global search procedure where one needs to instantiate numerous variations of silhouettes from the latent space. Instantiations should be compared to the silhouette in the currently captured frame. Then the closest silhouette's pose can be considered as the pose of the currently captured hand silhouette.

3) Mapping from Feature Space Y to Latent Space X: GPLVM finds a backward mapping from latent space X to feature space Y. For the real time application of the hand pose tracking system, a mapping from feature space to data space is required. Since this mapping is not

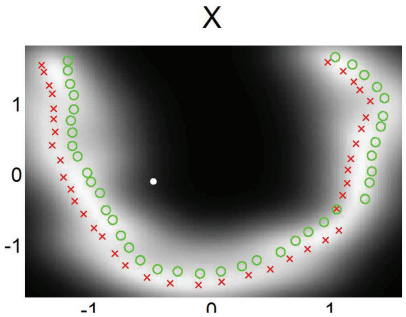


Fig. 10. Captured Manifold of Stereo Hand Silhouettes in 2D latent space X.

provided by the GPLVM, it is infeasible to generate all the possible X to Y mappings by observing the generated silhouette with the currently captured one. Therefore a mapping from Y to X is also required. This mapping will be used as an initial point of a local search algorithm in the latent space afterwards. An MLP with one hidden layer with 15 hidden neurons is used for learning the mapping from feature space to latent space.

4) Classification in Latent Space: 2-dimensional latent space has been found in a smooth fashion by GPLVM optimization. Therefore nearest neighbor matcher has been used in the latent space as a classifier without applying a local search algorithm. Ground truth angles of the hand poses are known. An exact pose match is looked for. Any divergence from the exact angles is considered as a classification error. For the synthetic environment prepared by poser a classification performance of 94% has been reached in 2D latent space.

D. Gesture and Expression Classifier

The gesture and expression recognition module is mainly used to give high level commands to the puppet. This is either achieved by performing hand gestures or with facial expressions. Hence, performers can initiate complicated animation sequences by performing a certain hand gesture, or they can change the appearance of the puppet by making certain facial expressions. For instance, a certain hand movement and posture can make the puppet jump or dance around, and performing a happy face can *make* the puppet happy via changes in textures or posture.

Since the output is an animation that is meant to be eye pleasing, discontinuities in the animation are not desirable. Therefore, there are no predetermined gestures, hand shapes or expressions that inform the system about the start or end of a gesture. This means that all gestures or expressions are performed continuously, with no indicators for separation. Thus, the gesture classification module also needs to *spot* known gestures or expressions in continuous streams.

1) Preprocessing: Each module provides a continuous stream of data consisting of different feature vectors. Pose estimation module uses skeleton parameters, motion tracking module uses 3D motion parameters of the hand, and the facial landmark tracking module uses landmark locations as feature vectors. Even though a generic sequence classifier such as a hidden Markov model (HMM) can be used to recognize patterns in each of these streams, the characteristics of the feature vectors are significantly different, and require separate preprocessing steps before using a generic recognition module.

Gestures defined with hand motions are scale invariant, and the starting point or the speed of the gesture is not important. Therefore, absolute coordinates of the hand location make little sense. In a preprocessing step, we find the relative motion of the hand in each frame, and then apply vector quantization to simplify calculations.

Changes in hand posture do not possess the characteristics of hand motion. As the hand posture is basically the rotation parameters of each joint in the hand skeleton, scaling and translation do not affect the resulting feature vectors. Therefore, the absolute parameters can directly be used. As there are more than 30 degrees of freedom associated with each hand, most of which are either unused or correlated, we also apply PCA to reduce the dimensionality. The reduction matrix is learned from the training data retrieved from Poser.

Facial expressions are very different, as they represent states as well as processes. Therefore, facial features should be used to recover both dynamic and static aspects of the face. Also, the absolute locations of landmarks is affected by the global motion of the head. We first estimate this global motion and reverse it to find local changes in face. Then we use the first derivative of the locations and apply PCA. Each static state of the face have the same observations this way, since each of them correspond to zero motion. By an intelligent choice of number of states and training data, we correctly represent each static state via the dynamic processes that lead to them.

2) Hidden Semi Markov Models: By far the most common method used for sequence classification is by using HMMs. HMMs model sequences with latent states, and the state durations implicitly via the self transition probabilities of each state. This leads to a geometric distribution of durations of each state. As the states model subsequences of gestures or expressions, modeling every duration with a geometric distribution can have undesirable effects. The graphical model of HMMs is given in Figure 11.

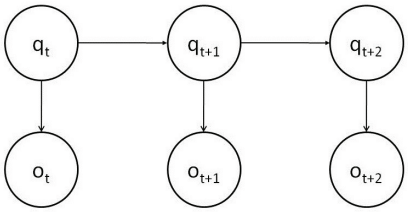


Fig. 11. Graphical model of a HMM.

HSMMs are extension of HMMs that can be thought of as graphical models consisting of *generalized states* emitting sequences, instead of states emitting a single observable [9]. For a general HSMM, there is no independence assumption for these emitted sequences and their durations. Also, the sequences emitted by the generalized states, i.e. the *segments* can have any arbitrary distribution. Different constraints on these distributions and assumptions of independence lead to different types of HSMMs. For instance, each segment can be thought of as produced from another HMM or state space model embedded in the generalized state, in which case the HSMM is known as a *segment model*. On the other hand, if the segment consists of a joint distribution of conditionally independent and identically distributed observations, the model becomes an explicit duration model [10]. Other variants include 2–vector HMM [11], duration dependent state transition model [12], inhomogeneous HMM [13], non–stationary HMM [14] and triplet Markov chains [15]. Detailed overview of HSMMs can be found in the tutorial by Yu [16].

HSMMs can be realized in the HMM framework, where each HMM state is replaced with a generalized or complex HMM state that consists of the HMM state and an integer valued random variable associated with that state, which keeps track of the remaining time. The state keeps producing observations as long as its duration variable is larger than zero. Hence, each state can emit a sequence of observations, the duration of which is determined by the length of time spent in that state. The corresponding graphical model is

depicted in Figure 12.

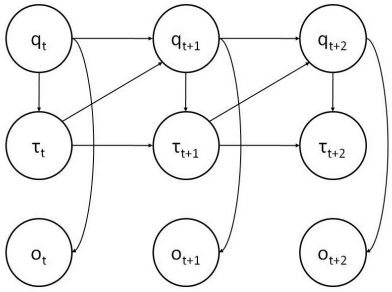


Fig. 12. Graphical model of a HSMM.

3) Continuous Recognition: HMMs and also HSMMs are quite simply applied to isolated gesture recognition problems. A separate model is trained for each class of pattern, and each candidate sample is evaluated against all the models. The class that corresponds to the model, which gives the highest likelihood is selected. This process is more complicated for continuous gesture recognition. A simple solution is to extract many candidate sequences that end at the current instance and have started at different instances. Using a threshold model for non–gestures (half-gestures, coarticulations, unintentional movements), one can determine the class of observed gestures. This is an inefficient method, as several sequences are evaluated at each frame.

In this work, we train a separate small HSMM for each gesture and expression, and then combine them to form a larger HSMM. Thus, a single (long enough) candidate sequence is evaluated with a single large model. Using Viterbi algorithm to estimate the optimal state sequence, one can determine the gesture class from the likeliest current state. The gestures and expressions are assumed to be independent. Therefore, the extracted candidate sequence can be rather short, as previous gestures have no effect on the current one.

At each frame, this module extracts the candidate sequences from each stream and applies the Viterbi algorithm for HSMMs. If the end of a pattern is recognized, a high level command is sent to the visualization module, triggering the corresponding animation sequence or posture.

E. Communication

Efficient communication between the components of the system is a crucial task. Reducing the response time helps the performers, since they receive feedback faster. To retain genericness and simplicity of the system, an XML based communication protocol is used. Visualization module accepts binary XML files from other modules, parses them and applies parameters directly to the ongoing animation in real–time. The XML-based protocol is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<handPuppet timeStamp="str" source="str">
    <paramset>
        <H rx="f" ry="f" rz="f" />
        <ER ry="f" rz="f" />
        <global tx="f" ty="f" ry="f" rz="f" />
    </paramset>
    <anim id="str" />
    <emo id="str" />
</handPuppet>
```

Here, keywords *timestamp* and *source* are used to identify, sort and prioritize messages received from modules. The *paramset* subtree holds the low level joint modification parameters. Each item in this subtree correspond to a single joint. As the system uses a rigid hierarchical skeleton, joints only need the rotation parameters. Translation and scaling are not allowed for joints. In order to move the entire skeleton, the keyword *global* is used, which also allows translation parameters. The *anim* keyword is used to trigger predefined animation sequences, which are bound to hand gestures. Likewise, the *emo* keyword is used to trigger predefined changes in appearance in the 3D model, which are triggered by detected facial expressions.

F. Visualization Module

Visual output of this project is an animated avatar controlled by the performers via several input methods. In order to let the users have an adequate control of this visual representation, we have utilized a skeleton based animation technique. The skeleton consists of 16 joints and accompanying bones. Using a graphical user interface, the user can create target poses for the skeleton as animation keyframes. Then it is possible to create an animation between these poses using quaternion-based spherical linear interpolation. These can be saved and then later be triggered by the modules upon recognition of certain gestures and expressions. The humanoid skeleton can be seen in Figure 13.

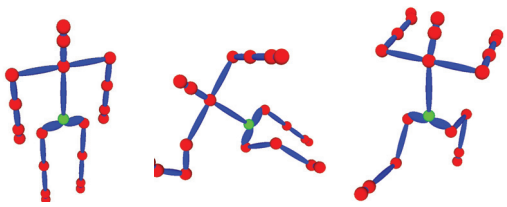


Fig. 13. Humanoid skeleton used for rigging and verification.

The skeleton has a humanoid form, but the actual 3D model does not need to be a humanoid model. The same skeleton can be bound to distinct models through rigging, which may include cartoonish creatures, or even objects with no real skeletons, such as trees or buildings.

To create a more realistic link between gesture inputs and animation output, an inverse kinematics based animation technique is implemented. This technique allows us to define end effector positions as well as particular rotations for each joint. In our context, an end effector refers to a hand, a foot or the head of the puppet. By setting goal positions for these end effectors, and applying constraints at each joint, we were able to produce a more realistic animation for the puppet.

For inverse kinematics, we have used the cyclic–coordinate descent algorithm. This algorithm starts the computation with the furthest joint away from the root. Then, each joint is traversed to minimize the difference between end effector and goal, optimally setting one joint at a time. With each joint update the end effector is also updated.

A generic model loader had also been implemented. This model loader accepts Autodesk FBX files as input, and associates model's geometric data with the underlying skeleton. When the user interacts with the skeleton, the spatial changes are automatically reflected to model's geometry, utilizing the association between model surface and skeletal bones.

IV. Conclusions

In this project, we developed a multimodal interface for digital puppetry. The design of the system allows manipulation of the low- and high level aspects of 3D hierarchical digital models in real–time. The hands and the face of multiple performers are tracked in order to

recognize their gestures and facial expressions. Each of the high and low level parameters estimated are mapped to kinematic parameters of digital puppets. The puppet is then animated accordingly, using several constraints and inverse kinematics.

Each module can run in separate workstations and communicate over the network using an XML based packet design. Also, each type of module can be used multiple times, and each such module can be associated with different aspects of the puppet. The methods employed do not depend on special hardware and do not require high computational power. Thus, the number of performers that can perform concurrently is only limited by the bandwidth of the visualization computer.

Each of the modules have been developed independently, and is shown to run in an efficient manner. Yet, the network protocol has not been adopted by every module, and the end result has not been demonstrated. When the modules can communicate with the visualization module, usability tests will be conducted and the interface will be improved accordingly. This is left as a future work.

V. Acknowledgments

This work is partially funded by the German Research Foundation (DFG) under Sonderforschungsbereich SFB 588 - Humanoid Robots - and by BMBF, German Federal Ministry of Education and Research as part of the GEMS programme. This work has also been supported by the Tübitak project 108E161.

References

[1] Son L. Phung, Abdesselam Bouzerdoum, and Douglas Chai, "Skin segmentation using color pixel classification: Analysis and comparison", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 148–154, 2005.

[2] Michael Isard and Andrew Blake, "Condensation - conditional density propagation for visual tracking", *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[3] B. Fasel and J. Luettin, "Automatic facial expression analysis: A survey", *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[4] M. Pantic and L. Rothkrantz, "Automatic analysis of facial expressions: the state of the art", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424–1445, 2000.

[5] İsmail Arı, "Facial feature tracking and expression recognition for sign language", Master's thesis, Boğaziçi University, 2008.

[6] S. Yildirim M. Bulut C.M. Lee A. Kazemzadeh S. Lee U. Neumann C. Busso, Z. Deng and S. Narayanan, "Analysis of emotion recognition using facial expressions, speech and multimodal information", *Proceedings of the 6th international conference on Multimodal interfaces - ICMI '04*, p. 205, 2004.

[7] Y. Wei, "Research on facial expression recognition and synthesis", Master's thesis, Nanjing University, 2009.

[8] D. Cooper T. Cootes, C. Taylor and J. Graham, "Active shape models-their training and application", *Computer vision and image understanding*, vol. 61, pp. 38–59, 1995.

[9] Kevin P. Murphy, "Hidden semi-markov models", Tech. Rep., 2002.

[10] J.D. Ferguson, "Variable duration models for speech", *Proceedings of the Symposium on the Application of Hidden Markov Models to Text and Speech*, pp. 143–179, 1980.

[11] Vikram Krishnamurthy, John B. Moore, and Shin-Ho Chung, "On hidden fractal model signal processing", *Signal Process.*, vol. 24, no. 2, pp. 177–192, 1991.

[12] S.V. Vaseghi, "Hidden markov models with duration-dependent state transition probabilities", *Electronics Letters*, vol. 27, no. 8, pp. 625–626, 1991.

[13] P. Ramesh and J.G. Wilpon, "Modeling state durations in hidden markov models for automatic speech recognition", *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1, pp. 381–384, 1992.

[14] Bongkee Sin and Jin H. Kim, "Nonstationary hidden markov model", *Signal Process.*, vol. 46, no. 1, pp. 31–46, 1995.

[15] Hulard C. Pieczynski, W. and T. Veit, "Triplet Markov chains in hidden signal restoration", in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2003, vol. 4885, pp. 58–68.

[16] Shun-Zheng Yu, "Hidden semi-markov models", *Artif. Intell.*, vol. 174, no. 2, pp. 215–243, 2010.

Lale Akarun is a professor of Computer Engineering in Boğaziçi University. Her research interests are face recognition and HCI. She has been a member of the FP6 projects Biosecure and SIMILAR, national projects on 3D Face Recognition and Sign Language Recognition . She currently has a joint project with Karlsruhe University on use of gestures in emergency management environments, and with University of Saint Petersburg on Info Kiosk for the Handicapped. She has actively participated in eNTERFACE, leading projects in eNTERFACE06 and eNTERFACE07, and organizing eNTERFACE07.

Dr. Rainer Stiefelhagen is a Professor at the Universität Karlsruhe (TH), where he is directing the research field on "Computer Vision for Human-Computer Interaction". He is also head of the research field "Perceptual User Interfaces" at the Fraunhofer Institut for Information and Data Processing (IITB) in Karlsruhe. His research focuses on the development of novel techniques for the visual and audio-visual perception of humans and their activities, in order to facilitate perceptive multimodal interfaces, humanoid robots and smart environments. In 2007, Dr. Stiefelhagen was awarded one of the currently five German Attract projects in the area of Computer Science funded by the Fraunhofer Gesellschaft. His work has been published in more than one hundred publications in journals and conferences. He has been a founder and Co-Chair of the CLEAR 2006 and 2007 workshops (Classification of Events, Activities and Relationships) and has been Program Committee member and co-organizer in many other conferences. Dr. Stiefelhagen received his Doctoral Degree in Engineering Sciences in 2002 from the Universität Karlsruhe (TH).

Hazim Ekenel is the head of "Facial Image Processing and Analysis" young investigator group at the Department of Computer Science in Karlsruhe Institute of Technology (KIT), Germany. He received his B.Sc. and M.Sc. degrees in Electrical and Electronic engineering from Boğaziçi University in 2001 and 2003, respectively, and Ph.D. degree in Computer Science from the University of Karlsruhe (TH) in 2009. He has been developing face recognition systems for smart environments, humanoid robots, and video analysis. He had been the task leader for face recognition in the European Computers in the Human Interaction Loop (CHIL) project and he organized face recognition evaluations within the CLEAR 2006, 2007 international evaluation campaigns. He has been responsible for face recognition in the German Humanoid Robots project. He is currently the task leader of face recognition in the French-German Quaero project. He has received the EBF European Biometric Research Award in 2008 for his contributions to the field of face recognition. In addition to the scientific work, many real-world systems have been developed based on his algorithm. With these systems, he received the Best Demo Award at the IEEE International Conference on Automatic Face and Gesture Recognition in 2008.

Cem Keskin is a Ph.D. candidate in the Computer Engineering Department at Boğaziçi University. He received his B.Sc. in Computer engineering and Physics from Boğaziçi University in 2003. He completed his M.Sc. in Computer engineering in the same university in 2006. His research interests include pattern recognition, computer vision, human computer interfaces and hand gesture recognition. He also took part in or contributed to several projects and applications, such as 3D reconstruction of buildings from multiple images, realistic and automatic coloring of buildings on real images, machine learning for genetic research, hand gesture based emergency control rooms, hand gesture based video games, synthesis of musical score from body movements and software algorithms for advanced computer graphics. He has a 2nd degree in the best student paper contest in SIU 2003. The title of his Ph.D. thesis reads "Generative vs. discriminative models for analysis and synthesis of sequential data".

Lukas Rybok graduated from the University of Karlsruhe (TH), Germany in 2008 with a Diploma degree in Computer Science. He is currently a Phd candidate at the Karlsruhe Institute of Technology (KIT) working in the field of human activity recognition under the supervision of Prof. Rainer Stiefelhagen. His research interests include computer vision, pattern recognition and machine learning.

Furkan Kıraç received the B. Eng. degree on Mechanical Engineering in 2000 and M. Sci. degree on Systems and Control Engineering in 2002 from the Boğaziçi University. He is currently a PhD candidate in Computer Engineering in Boğaziçi University. Since 2000, he has been actively working on computer vision based software design and has been the founder of a computer vision firm named Proksima which developed license plate recognition software since 2002. He has received a number of awards for his programming skills from TÜBTAK (The Scientific and Technological Research Council of Turkey). He has been given the 3rd degree in National Science Competition on Computer Science in Turkey, both in 1994 and 1995. He also has 1st and 2nd degrees in Regional Science Competitions of Marmara Territory in 1994 and 1995 respectively. Due to his performance in the national competitions he has represented Turkey in "London International Youth Science Forum '95" on computer science. He also has a 2nd degree in Best Paper Contest in SIU 2005 National Conference in Turkey. He is currently continuing his PhD research on "Human Motion Understanding" in Computer Engineering Department of Boğaziçi University, Turkey.

İsmail Arı is a PhD candidate and teaching assistant in the Department of Computer Engineering in Boğaziçi University. His research interests include facial expression recognition, computer vision, and machine learning. Arı has an MS from the same department. Contact him at ismailar@boun.edu.tr.

Mustafa Tolga Eren is currently a PhD candidate in Computer Science and Engineering and a research assistant at Computer Graphics Laboratory in Sabancı University. He received his B.S. degree on Computer Science and Engineering Program from Sabancı University in 2006. His research interests include augmented reality, virtual environments and physically based simulations and animation.

An Audio-Visual Speech Recognition System with Live Inputs

Ibrahim Saygin Topkaya, Mustafa Berkay Yilmaz, Umut Sen, Alexey Tarasov, Hakan Erdogan

Abstract— **This project aims to build an audio-visual speech recognizer application which responds to preset commands and performs corresponding operations accordingly. The system, while in action, runs in three steps; data acquisition, feature extraction/processing and recognition. When triggered, it records a short segment of user's spoken utterance through video and audio and finds a region of interest in the video frame. Finally it extracts useful features from the recorded channels' data and tries to decode the spoken utterance. We aim to show how visual channel and derived information from it can be used to support information acquired through audio channel and give information about how frame asynchrony between two channels can be handled in a real time application. We also propose to use visual tandem features obtained through classifiers as additional streams to improve the audio-visual recognition accuracy.**

I. Introduction

Speech recognition from audio is a mature technology where usually hidden Markov models (HMM) are used to model sequential audio features with a hidden state machine that has Markovian transitions and emission likelihoods generally modeled with Gaussian mixtures [1], [2]. However under conditions where audio information is not enough (e.g. due to noise), supporting audio channel with visual information is a technique that is used to improve recognition accuracy [3].

HMM's generative modeling of the observation data can be supported by discriminative classifiers using a technique called the tandem approach [4]. This approach is used to improve recognition accuracy by utilizing the outputs of the discriminative first level classifiers (e.g. Support Vector Machines (SVM) [5] or Neural Networks (NN) [6] as observations in the hidden Markov model.

In this work we investigate practical issues for building such a visually supported speech recognizer, also propose to use tandem classifiers for visual data, and use them alongside with regular streams to improve audio-visual recognition accuracy. The recogniser handles these different kinds of data in separate streams, resulting in a multi-stream HMM (MSHMM) [7]. To demonstrate the recognition process, we build an application that responds to spoken utterances and recognizes the spoken command to perform basic tasks on the computer. While running, the application records a short spoken utterance and finds visual lip region of interest (ROI) automatically. After the data capturing and ROI extraction is complete, it extracts useful features from both audio and visual ROI data and applies tandem classifiers to the visual features. Then using all of these observation features and derived tandem features, tries to deduce the spoken command and performs the preset operations according to the decoded command. Also, extracted features are preferably saved as feature files to use in model training and offline experiments.

The software uses HTK compatible models for MSHMM recognition [8], so the user can train a custom language model with HTK toolkit applications by using the saved feature files. The accuracy of the presented system is tested with offline experiments by using the feature files that are generated by the developed application. The results of these experiments give an overall information about the accuracy of the audio-visual speech recogniser.

I. S. Topkaya, M. B. Yilmaz, U. Sen and H. Erdogan are with Vision and Pattern Analysis Laboratory, Sabanci University, Turkey. A.Tarasov is with Digital Media Centre, Dublin Institute of Technology, Ireland.

The report is presented parallel to the working modules of the application; in the second section, an overall information about the whole system is presented. Following that, data acquisition modules for both audio and video channels are explained, together with visual ROI extraction module and feature extraction modules. Although implementation and experiments of tandem approach did not finish by the end of workshop, fourth section gives information about the tandem approach and how it can be integrated into the recognition process. Fifth section gives the details of the multi-stream model that performs the actual recognition, and finally results are presented.

II. Overall Application Architecture

The application is aimed to run on multiple platforms and use free components. It is aimed to be a standalone application and perform the recognition task out of the box. The whole components are developed with C++ language, where Qt [9] is selected as the main development platform and base environment. All external libraries that are used are free and available on 3 major platforms available for personal computers–Windows, Linux and MacOS X.

The main focus of application is performing live recognition using audio and visual features. So to run live, it needs pre-trained hidden Markov models for main recognition. Although training of these classifiers are out of the scope of the application, the application also provides material to train these classifiers such that in live recognition mode it reads classifier information from external files, however also it has the option to only extract features and save feature files to train those classifiers with external tools–details of which are explained in following sections.

Whether live recognition is performed or only feature files are generated, the capturing process of the application is common; when triggered it records a limited time of audio-visual data in two parallel threads, finds a mouth region and extracts useful features from captured data. The main distinction is; during live recognition it records a short (nearly one-second) data and tries to recognize the spoken word during that time. However during feature file generation, it allows user to speak more than one word since the user can train the HMM with files containing more than one word.

After visual ROI and audio-visual features are extracted, the application continues in two different ways according to user selection:

- If the user selects saving feature files, the application writes the extracted features in two different files (one for each channel) and ends processing
- If the user selects recognition, using pre-trained hidden Markov models, the application decodes the spoken word and performs the corresponding command

The details of all these steps are given in the following sections.

III. Data Acquisition and Observation Features

A. Structure of The Data Buffer

Initially we were planning to use one single complex buffer for captured data and derived features. Although we have decided to move on to two separate buffers because of channel asynchrony, the main idea remains the same, so here we present information about the whole buffer model.

The buffer is actually an array of a complex class, where each array element corresponds to a feature frame that will be processed by the MSHMM. For visual features, this corresponds to a frame captured through camera and usually transformed to another space (e.g. DCT [10]). For audio features, one frame corresponds to a time slice of the audio signal, usually partially overlapping with neighbouring frames and transformed to another space (e.g. MFCC [11]).

The visual buffer consists of visual elements; which are captured frames through the camera and corresponding feature frames, so the structure of a single visual buffer frame is:

- **Visual Frame:** Actual frame grabbed from camera
- **Frame ROI:** Lip region that is extracted from the visual frame
- **DCT Frame:** Lip region image converted to DCT space
- **DCT Features:** First few high energy features, stacked as a 1D array

As soon as the capturing begins, visual frame elements of the buffer are filled with captured frames. After the capturing ends, the module that extracts the lip region runs and extracts the lip ROI on the whole captured scene. After the lip ROI is extracted, for each frame the extracted ROI image is transformed to the DCT space. Finally, first few high energy horizontal and vertical DCT coefficients are stacked as a one dimensional array to be used in recognition as the visual observation features.

The audio buffer consists of audio elements, which are captured frames and corresponding feature frames, so the structure of a single buffer frame is:

- **Audio Frame:** An analysis window of captured audio samples
- **MFCC Features:** Audio features extracted from the analysis window

Similarly as soon as the audio capture begins, audio frame elements of the buffer are filled with captured samples. After the capturing ends, the module that extracts the MFCC features runs and for each frame, derives MFCC coefficients of the frame.

The capturing process runs in real-time in two different threads– one for audio and the other for visual features. The multi-threading solution is performed with Qt's threading capabilities. Since two channels' buffers are independent during capturing, there is no need of thread synchronization during capturing.

B. Audio Data Acquisition

For acquiring audio data, PortAudio library is preferred [12], because of its ease of use and multi-platform availability. Before data capturing begins, PortAudio is configured for capturing single channel (mono) audio input with a sampling frequency of 16 kHz. To make audio feature extraction phase easier, captured audio packets are written onto the audio buffer with overlapping windows. To be compatible with the parameters used for audio feature extraction, audio packets are captured having length of 80 samples for each packet.

C. Audio Feature Extraction

1) Mel-Frequency Cepstral Coefficients: In the tasks of pattern recognition the reduction of the number of features associated with an object is needed. The main reason is that it can make the training process of the classifier much shorter and the performance, as well as the robustness, of the resulting classifier higher. One of the algorithms that is widely used to compress signals is the Discrete Fourier Transformation (DFT) [13] that calculates the spectrum of a signal. Then some number of the first amplitude values from a spectrum are used as a features instead of the original signal data, and omitting the rest. But in the field of sound recognition the DFT has a significant drawback—it can not model the perception of a sound by human properly.

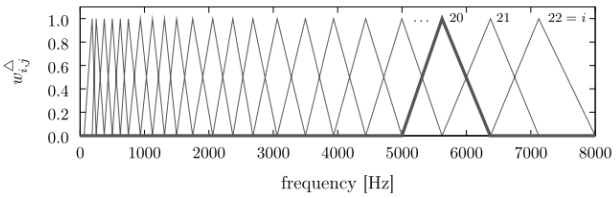


Fig. 1. An example Mel-filterbank consisting of 22 triangular filters [14]

The problem is that the spectral resolution of human auditory system is not linear along the frequency axis. For instance, a human can easily discriminate between a sound of 150 Hz and 200 Hz, but not between 2000 and 2050 Hz. To resolve this problem, a special Mel-scale was proposed [11]. The spectral resolution mentioned earlier is linear along its axis. The frequency f measured in Hz could be converted to the according m value on the Mel-scale using a formula

$$m = 2595 \cdot \log_{10}\left(\frac{f}{700} + 1\right) \tag{1}$$

Mel-frequency cepstral coefficients (MFCCs) make use of this scale to perform a compression of a sound signal. To derive them, first it is necessary to calculate a spectrum of the sound signal performing the DFT. Then a special set of overlapping triangular filters called a Mel-filterbank is applied to the spectrum. An example of it is shown in Figure 1. In order to use it, frequency range that it will cover should be defined, in this example it is from 8 Hz to 8kHz. After it the number of filters has to be specified—it is 22 in the given case. Their centers are equally distributed in the frequency domain, but the application of formula (1) to these values make them more dense in the lower part of the Mel-scale. After it the energy c_i^{Mel} for each of the i-th filter ($i = 1..F$, where F is a number of filters in a Mel-filterbank) is calculated:

$$c_i^{Mel} = \sum_{j=1}^{N} w_{i,j}^{\Delta} \cdot |c_j^{DFT}|^2, \tag{2}$$

where N is the number of frequency bins in the initial sound spectrum received by DFT, $w_{i,j}^{\Delta}$ is the value of the i-th Mel-filterbank filter for the j-th frequency bin and c_j^{DFT} is the spectral value at the j-th frequency bin.

After filtering, logarithms of $c_i^{Mel}, (i = 1..F)$ are taken and a discrete cosine transform (DCT) is applied to the resulting values. The amplitudes of the lower order DCT coefficients starting from one are the MFCC values.

In general, only twelve first MFCC coefficients are used together with the energy coefficient, which is the logarithm of the energy (sum of squares) of the audio samples in the frame.

Using the Mel-scale allows MFCCs to represent only *what* has been spoken, throwing away the information about *how* it has been pronounced. This is the main reason why MFCCs are widely and successfully used in the field of audio speech recognition.

2) Implementation of MFCC Extraction: For MFCC features, overlapping windows having 400 samples are used and distance between consecutive windows is taken as 160 samples. This resulted in 100 windows for one second thus 100 audio feature frames per second.

As mentioned, raw audio data captured with PortAudio is written onto the audio buffer with overlapping windows, where each buffer frame contains 400 samples, thus 5 captured packets. Also since the

Fig. 2. Example face and nose region extraction results

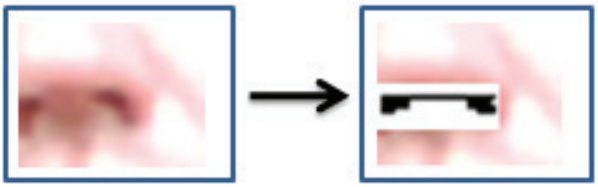


Fig. 3. Example nose thresholding result

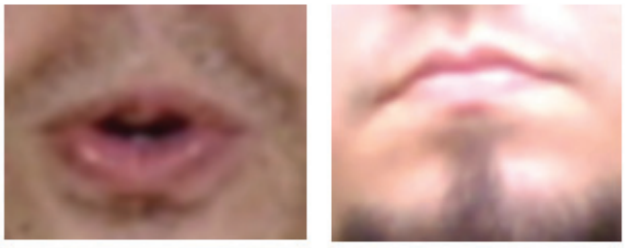


Fig. 4. Example lip regions

windows are overlapping, each packet is written more than once to the buffer on the neighboring buffer frames. Although this means storing redundant information in the buffer and increasing the buffer size, storing so made the MFCC extraction phase much easier.

For MFCC extraction, libXtract library [15] is preferred for similar reasons with PortAudio. One buffer frame corresponds to one audio feature window and after a buffer frame is filled with raw audio data, using libXtract functions short-time Fourier transform of the window is calculated, followed by the calculation of the MFC coefficients. So to calculate the audio features of one buffer frame, only information needed is already available in the same frame. Following general convention only first 13 MFC coefficients are used as the observation vector of the audio stream.

D. Visual Data Acquisition

For acquiring visual data OpenCV library [16] is preferred, which too is available on multiple platforms. OpenCV is configured to capture visual frames continuously through the chosen camera. Each captured frame is written onto the visual buffer to be handled with other related modules.

To extract visual region of interest (ROI), we work with a strategy to use the nose region as a pivot to obtain the lip region. The reason for this choice is that the lip region is more varying in appearance as compared to the nose region and since the nose region does not change as much as the lip region, we track the nose region to derive the lip ROI from it.

E. Visual ROI Extraction

1) Nose Region Extraction: First step of lip region extraction is to find the nose region. A video frame is first converted to gray-level. Then face and nose regions are extracted sequentially, using the Viola-Jones haar-like feature cascades [17]. Middle part of the face is considered when detecting the nose to eliminate potential incorrect detections. Example extraction results are shown in Figure 2.

2) Nostril Detection: Once the nose region is extracted, nostrils are detected inside that area. Nostrils determine the upper bound of mouth region. Nostrils are found in nose region by thresholding. Best threshold is found looking at nose region histogram. Ideal threshold should leave out only a small percentage of the whole nose region. Both ends of the darkest area in nose region are accepted as approximate nostrils. Center of nostrils becomes the upper boundary of mouth region. An example thresholding result is shown Figure 3.

3) Lip Region Extraction: Left and right lip corners are determined from face region using a basic assumption: Left corner is assumed to be located at $w/4$ horizontally, where w is the width of face region. Right lip corner is assumed to be located at $3w/4$, similarly. With the knowledge of top, left and right ends of the lip region, bottom end is found by a square lip region assumption. Two example lip regions are shown in Figure 4.

4) Lip Region Tracking: Once the lip region is extracted, it is not necessary to find it again and again in every consecutive frame. Instead, the region of interest is tracked. The idea behind the tracking is based on the assumption that, nose region does'nt change its appearance within the frames. Correlation of the nose with the face is found in every frame. The point giving the highest peak is compared with the previous highest peak. Difference of the locations gives an estimate of head motion, assuming no head rotation. Average translation of several (usually 10) previous frames is used to smooth the estimated motion.

5) Lip Region Reinitialization: During the video, sometimes it is not possible to track the lip region correctly for the whole talk. Even the lip region extracted in first frame may not be correct. The tracking process is reset and everything is found again if one of the following conditions holds:

- Total displacement from the latest reinitialization exceeds a certain threshold. Because of big translation, tracker may not keep up with the lip region.
- Within some number of frames from the latest reinitalization. Scene may totally change during the tracking or the found lip region from the latest reinitialization may be incorrect.

F. Visual Feature Extraction

1) Discrete Cosine Transform: The Discrete Fourier Transformation (DFT) is used in many application domains to compress a discrete signal $g(u)$ having M values, representing it as a spectrum $G(m), (m = 0, \ldots, M-1)$, i.e. as a sum of cosine and sine functions:

$$G(m) = \frac{1}{\sqrt{M}} \sum_{u=0}^{M-1} g(u) \left[\cos\left(2\pi \frac{mu}{M}\right) - i \cdot \sin\left(2\pi \frac{mu}{M}\right) \right] \quad (3)$$

The DFT is designed for processing complex-valued signals, but in the case of real values faster algorithms could be used. The discrete cosine transform (DCT) [10] is an example of such an algorithm, that is used frequently for compressing images and video data. It calculates a spectrum of a signal as follows:

$$G(m) = \sqrt{\frac{2}{M}} \cdot \sum_{m=0}^{M-1} \left[g(u) \cdot c_m \cdot \cos\left(\pi \frac{m(2u+1)}{2M}\right) \right] \quad (4)$$

with

$$c_m = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } m = 0 \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

Let us denote a basis function of the DCT as

$$\mathbf{D}_m^M(u) = \cos\left(\pi \frac{m(2u+1)}{2M}\right), \quad (6)$$

then the two-dimensional form of the DCT could be expressed as

$$G(m,n) = \frac{2c_m c_n}{\sqrt{MN}} \cdot \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \left[g(u,v) \cdot \mathbf{D}_m^M(u) \cdot \mathbf{D}_n^N(v) \right] \quad (7)$$

where c_m and c_n has the same meaning as in (5), N is the number of signal values in the second dimension and $n = 0, \ldots, N-1$.

In such formulation the DCT could be used for images, that in fact are 2D signals. Using DCT features allows compression and compact representation similar to MFCC features; a low number of DCT features are enough to represent most variety in the image, thus omitting a high number of DCT features is enough to represent the source image.

2) Implementation of DCT Extraction: For DCT features, extracted ROI of the lip region in grayscale format is used as source for DCT conversion which is performed again with OpenCV library. Obtained DCT image has the same size of pixels as the ROI image, however similar to MFCC of audio features, only first five horizontal and vertical DCT coefficients are used as visual features. These twenty-five features are packed as a one dimensional array and are used as the observation vector of the visual stream.

G. Synchrony of Audio and Visual Data

When offline experiments are performed for audio-visual speech recognition, usually recorded video files are used where a video file generally has a constant frame per second (FPS) rate for (e.g. 25 FPS for PAL DVDs) visual data. Also the number of windows used in one second during MFCC feature extraction gives the number of audio feature frames per one second (e.g. usually 100 FPS for speech applications).

Since audio and visual data are processed in parallel with MSHMMs, it is needed to set the rate of both data equal which is usually done by upsampling the visual data. To upsample a video file aimed for watching, one can do a motion analysis based interpolation of video frames which takes movement of objects in consideration so that there is as little interpolation artifact as possible on the output video. However for our case where the DCT features of frames are used for recognition, a simpler interpolation scheme which only interpolates intermediate frames is enough.

For an offline experiment with constant video FPS, there is a constant upsampling ratio which is applied to the whole dataset used in the experiment. However one issue we have noticed during live acquisition is that, when a webcam with a moderate hardware is used, such a constant FPS and thus a constant upsampling ratio is not obtained. We first noticed effects of this asynchrony when we assumed such a constant ratio between audio and video channels and upsampled visual feature frames accordingly and got very poor results in offline experiments. In those experiments, when the visual features are involved even in a small weight, the recognition ratio decreased to zero dramatically. Although visual features are known to give lower accuracy than audio features this dramatic decrease to zero is out of the bounds of this evident fact.

After noticing this, we have developed a dynamic frame resampling scheme to overcome this problem. Since the videos are obtained in small chunks, the number of frames that will be sent to the MSHMM is decided as the number of audio feature FPS, since it is nearly 4 between 5 times higher than the number of visual FPS. Also, many libraries contain functions for image/matrix/vector resizing algorithms so we have built our resampling scenario using these algorithms.

Let D be total number of DCT features, N_v be total number of captured video frames and N_a be total number of captured audio frames. For every DCT feature, we first concatenate N_v number of same feature from each video frame and resize this vector of size N_v to N_a using linear interpolation. For interpolation, again we benefit from the OpenCV library which includes image/vector interpolation functions. This procedure is repeated for each DCT feature, so finally the number of visual and audio feature frames become equal. After that both feature frame sequences are taken as two different streams of the MSHMM. The pseudocode for the developed scheme is at Algorithm 1.

Algorithm 1 Psuedocode for visual feature upsampling

```
D=Total number of DCT features
V=Video Buffer
AV=Audio-Visual Buffer
N_v=Total number of frames in video buffer
N_a=Total number of frames in audio buffer
for d=1 to D do
    d_all={ }
    for v=1 to N_v do
        concatenate d_all with d of V[v]
    end for
    resize d_all from N_v to N_a by interpolation
    for a=1 to N_a do
        d of AV[a]=a of d_all
    end for
end for
```

IV. Tandem Features for Speech Recognition

Tandem approach is a well known technique especially in speech recognition from audio. The approach proposes using posterior probabilities of a classifier, rather than direct observations, as a feature vector in the HMM classifier. The idea is performed by adding a classifier layer after feature extraction. The class definition for the tandem classifier can be chosen parallel to the HMM, such that each class can be one of words, sub-words, phones, monophone states or context-dependent phone states.

For example, consider a monophone HMM model for single word recognition that is trained to recognize ten words, around twenty phones (depending on the language) and a total of sixty monophone states (three states for each phone), so the tandem classifier may be trained to discriminate one of these units of HMM.

The outputs of a classifier (e.g. posterior probabilities / margin distances for an SVM or output layer values for a NN) are then considered in the HMM model as observation vectors. Usually the values are directly used, so for a tandem classifier trained for a number of C classes, HMM observations are vectors of length C.

A. Using Multiple Tandem Streams in AVSR

Being originally proposed for audio-only speech recognition, the tandem idea can be used for video based features as well by applying the same process to extracted video features. In this work, we propose using the tandem approach for video data, using multiple classifier outputs in addition to regular observation features. Although the

implementation and experimentation of tandem classifiers did not finish by the end of workshop, for completeness and guide future research we present information on this section by referring our previous research [18]. For this approach, we propose to model all streams of data using a MSHMM where each stream comes from different sources; one for audio features, one for video features and two features extracted from tandem classifiers. A block diagram of the proposed system is shown in Figure 5.

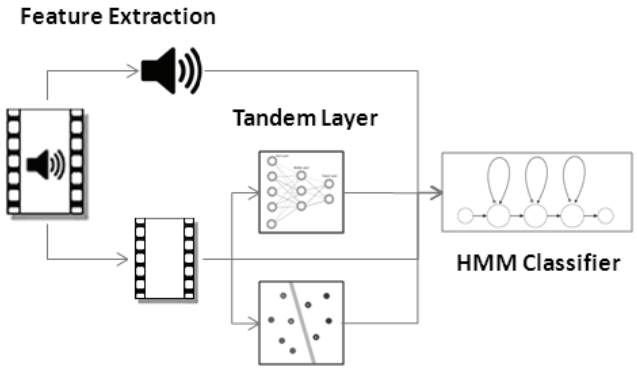


Fig. 5. Incorporating tandem streams into MSHMM.

B. Support Vector Machines as a Tandem Classifier

For the above mentioned tandem approach, we need a classifier to obtain the posterior probabilities and a convenient classifier is the support vector machine (SVM). It is used frequently in the literature lately for classification and regression problems. SVM works with the idea of modelling the boundary and constructing a classifier with maximum margin. An SVM constructs a max-margin hyperplane in a high dimensional feature space (with the help of the kernel trick).

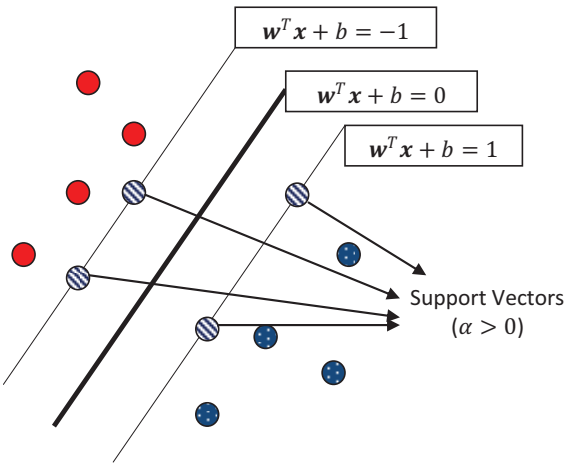


Fig. 6. An example to the hyperplane and support vectors in 2D

Given binary training data $(x_i \in R^n, i = 1, \ldots, l)$ and label vector $y \in R^l (y_i \in \{1, -1\})$, SVM solves the following problem:

$$\min_{w\, b, \xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

such that

$$y_i(\mathbf{w}^T\varphi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, i = 1, \ldots, l.$$

This optimization is equal to the following in the dual form:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}$$

such that

$$\mathbf{y}^T\boldsymbol{\alpha} = 0,$$

$$0 \leq \alpha_i \leq C, i = 1, \ldots, l.$$

where $\mathbf{e}$ is a vector composed of ones, $C > 0$ is upper bound, Q is an $l \times l$ matrix whose elements are dot products of data in the high dimension ($Q_{ij} = y_i y_j K(\mathbf{x}_i\mathbf{x}_j)$). K is the kernel function ($K(\mathbf{x}_i\mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$). Here, training data is projected to the higher dimension by the φ function.

The support vectors are found after training. In the test phase, the location of the data instances with respect to boundary is found using support vector machines with the formula below:

$$f(\mathbf{x}) = \sum_{i=1}^{l}\alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b.$$

This method is obtained for binary classification problems but can also be used for multi-class problems, for which one-against-one approach is used [19].

The f function above outputs the distance of the data instance to the margin, but for tandem approach we need posterior probabilities. For obtaining the posterior probabilities, an approach in [19] can be used.

C. Tandem Approach using Neural Networks

Another alternative for the base classifier in tandem approach is the artificial neural network (ANN). In Figure 7, an example of ANN is illustrated.

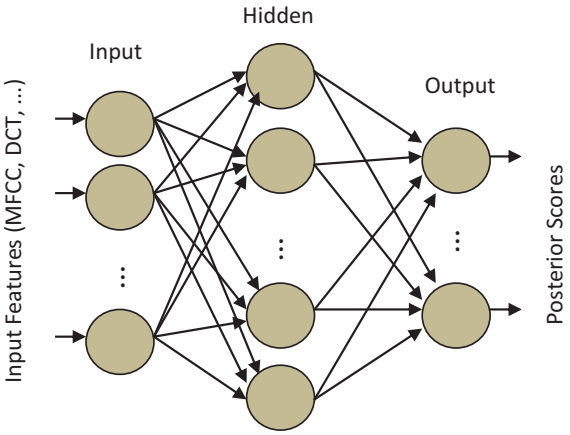


Fig. 7. An ANN with one hidden hidden layer

ANNs have at least 3 layers; more layers can be added as hidden layers and this results in more complex classifiers. ANN is modelled similar to the working mechanism of neurons in the brain and is a complicated layered network which combines simple basis classifiers. Each node is a composition of earlier layer values $g_i(\mathbf{x})$ and usually this composition is a nonlinear weighted sum:

$$f(\mathbf{x}) = K(\sum_i w_i g_i(x))$$

where K is the activation function and w_i is the weight of i^{th} component. There are weights for each connection between the nodes and each node can be thought as a simple classifier. Training the network means 'learning the weights' and the most popular algorithm is "back propagation". This algorithm aims to adjust the weights

for reduction of the error, which is actually a cost function, at each iteration and the weights are initialized randomly.

D. Implementation of Tandem Layer in The Application

After visual features are extracted by the methods explained in Section III the feature array is used as an input to the trained classifiers. For this step, OpenCV's machine learning classes are used which allow to incorporate both SVM and NN classifiers into the application.

Since the application also allows to collect feature files for model training, they can also be used to train the tandem classifiers. So for testing the application and the proposed methods, we first collect a small number of feature files of the sequences containing the spoken words. With a small application that uses OpenCV, we train two different tandem classifiers one using NN and one using SVM algorithms. We also split the training set into subsets (called tapes) and use stacked generalization with cross-validation to train the tandem classifiers and extract posterior probabilities. First training subsets are used as training data and last subset is used for testing in offline experiments. Each classifier in training set is trained using the whole training set excluding one subset and tandem features of the excluded subset is extracted using that trained classifier. Finally a classifier is trained using all training set and is used to extract features of the test data.

For the recorded feature files, the class of each frame is determined using alignment done from an HMM trained on only audio features, since clean audio is the most reliable data and we use it as a baseline for our model. We take phones as classes; since using words would result in a small number of complex classes. On the contrary using phone states would result in too many classes. So tandem classifiers discriminate each phoneme class, and generate feature vectors of length equal to the number of phonemes, where each dimension corresponds to the output of the classifier for each class.

For the future implementation, while the application is running for live recognition, extracted visual features are to be used as input for the trained models. With OpenCV methods, the classifiers generate posterior probability for each class, so for each of the two classifiers two different feature vectors are obtained, where length of each vector is equal to the number of phonemes. These two vectors for each frame are appended as additional members to the buffer elements, resulting in the final form of the buffer. In this form, where audio and upsampled video are combined also supported with tandem elements, each buffer element has the following structure:

1) **Audio Frame:** A window of captured audio samples
2) **MFCC Features:** First few lower order DCT coefficients derived from the audio frame
3) **Visual Frame:** Actual frame grabbed from camera
4) **Frame ROI:** Lip region that is extracted from the visual frame
5) **DCT Frame:** Lip region image converted to DCT space
6) **DCT Features:** First few lower order DCT coefficients, stacked as a 1D array
7) **NN Tandem Features:** Posterior probability for each class, generated by NN classifier
8) **SVM Tandem Features:** Posterior probability for each class, generated by SVM classifier

During recognition, elements 2, 6, 7 and 8 of the buffer are to be used as four static features for the streams of the MSHMM. We add dynamic features to the static features as well. We concatenate Δ and $\Delta\Delta$ coefficients for the audio features. and the Δ features for the video and video tandem features. Currently the application does audio-visual recognition without tandem features, so only first two of those are used.

TABLE I
BEST RESULTS USING TANDEM MSHMM.

SNR	Audio	Video	AudioVisual	AV+Tandem
Clean	100	36.67	100	**100**
20	99.17	36.67	100	**100**
15	93.61	36.67	96.67	**96.67**
10	74.44	36.67	81.67	**85.00**
5	37.50	36.67	54.44	**62.50**
0	11.39	36.67	36.67	**52.78**
-5	9.44	36.67	36.67	**46.39**
-10	6.11	36.67	36.67	**45.56**
-15	2.78	36.67	36.67	**45.56**
-20	6.94	36.67	36.67	**45.56**

To give an information about the improvement achieved with tandem features, we give here the results that are done on offline files. Although the experiment structure is different from the application since data is extracted from recorded video files, it can give information about the proposed approach. The results are tested with M2VTS database, where 10 digits are spoken. Four tapes of the database are used for training and last one for recognition where noise in different levels are added to the test tape, results of which can be seen on Table I. The results clearly show that when audio data is insufficient, incorporating visual data improves accuracy, and supporting it with tandem features improves more.

V. RECOGNITION WITH MSHMM

A. Multi-Stream Hidden Markov Models

Hidden Markov Models (HMMs) [1] are models that represent a signal as a random process in which input variables change in time. So they make it possible to use information about previous observations in the classification process.

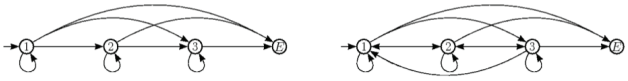


Fig. 8. Examples of 3-state HMM: left-to-right (left) and ergodic (right) [20]

HMMs are very similar to discrete Markov processes, but with a difference that states are not observable. The decision about the topology should be taken before the actual learning process happens. Two most popular options are shown in Figure 8.

As stated above, HMM states are unobservable in the observations so the change in the states is inferred from the change in the observations. The probability of observing a value in a state is estimated from the training set. If these variables are continuous, Gaussian Mixture Models [2] are usually used for describing the conditional probability density functions.

The most commonly used approach is to train one HMM per each class that has to be recognized. When the input signal is being recognized, probabilities that it could be generated by each of HMM are calculated i.e. we are identifying how close it is to each of the classes.

Let us consider the mathematical formulation of the classification task using HMMs. Let $S = \{S_1, S_2, \ldots, S_N\}$ be the set of all N states in one model/class. Let $p_j(x)$, $j = 1..N$ be the probability of getting an observation x (a feature value or a vector as well) in the j-th state. Let the vector of initial state probabilities be denoted as $\Pi = (\pi_1, \pi_2, \ldots, \pi_N)$, where π_j, $j = 1..N$ is the probability that the HMM will start in the j-th state. Let an N x N matrix $\mathbf{A}$ contain state transition probabilities, where element a_{rs}, $r, s = 1..N$ denote

a probability that the HMM will move to the s-th state from the r-th state.

If the sequence of states $\mathbf{Q} = (q_1, q_2, \ldots, q_T)$ is known, the probability of seeing an observation sequence $\mathbf{o} = (o_1, o_2, \ldots, o_T)$ on the HMM can be written as follows:

$$p(\mathbf{O}|\mathbf{Q}) = p_{q_1}(o_1)p_{q_2}(o_2)\ldots p_{q_T}(o_T) = \prod_{t=1}^{N} p_{q_t}(o_t) \quad (8)$$

The main problem with using (8) is that the state sequence is hidden. The probability of the state sequence $\mathbf{Q}$ in the model is:

$$p(\mathbf{Q}) = \pi_{q_1} a_{q_1,q_2} a_{q_2,q_3} \ldots a_{q_{T-1},q_T} \quad (9)$$

Using (8) and (9) joint probability can be formulated:

$$p(\mathbf{O}, \mathbf{Q}) = p(\mathbf{O}|\mathbf{Q})p(\mathbf{Q}) \quad (10)$$

Then the probability of seeing a particular observation sequence using (10) can be calculated as

$$p(\mathbf{O}) = \sum_{all\ possible\ \mathbf{Q}} p(\mathbf{O}, \mathbf{Q}) \quad (11)$$

Although there is a huge amount of possible $\mathbf{Q}$ sequences in formula (11) , this probability can be calculated using special forward-backward procedure.

The decoding task is performed as finding the underlying state sequence for a given observation sequence using the Viterbi algorithm [1].

HMMs are widely used as a method for audio speech recognition, however, they experience some problems while dealing with multiple streams, for instance, with video and audio data, together. The most straightforward way to combine the information from two streams is to put all the features into a single vector, but this approach treats all streams equal.

In a multi stream hidden Markov model [7], multiple streams of observations are handled in calculating the emission probabilities of the HMM model. Given a multi-stream observation sequence $(o_1, o_2, \ldots, o_T)$, we assume that each observation is a concatenation of multiple vector sources $o_t = [o_t^1, \ldots, o_t^S]$, where S is the number of modeled streams. The emission probability for a state q_t is calculated by:

$$p(o_t|q_t) = \prod_{i=1}^{S} p(o_t^i|q_t)^{\lambda_i}. \quad (12)$$

Here λ_i are the stream weights. For continuous valued observations (which is the case in speech recognition) each stream is usually separately modeled with a Gaussian mixture model. This is almost equivalent to assuming that each stream is independent in terms of emission probabilities when for the weighting factors λ_i are all one.

B. Training The Model For Recognition

The proposed MSHMM consists of four streams; (1) audio, (2) visual, (3) visual tandem using SVM classifier and (4) visual tandem using NN classifier, where currently only first two are handled by the application. The contribution of each stream to the decoding process differs on each experiment; we examine different stream weights between 0 and 1, in steps of 0.125 however while training we use stream weights of one. Since recognition is performed with HTK library functions in the application, the models that are needed while the software is being actively used during recognition can be any model that is trained by HTK.

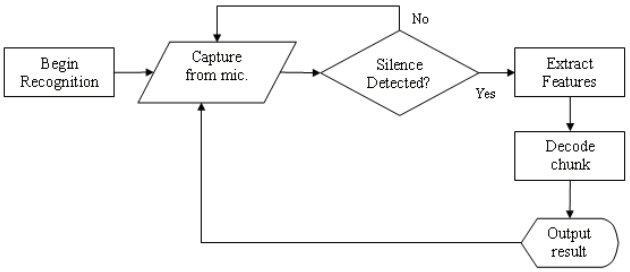


Fig. 9. Flow diagram of regular HVite running

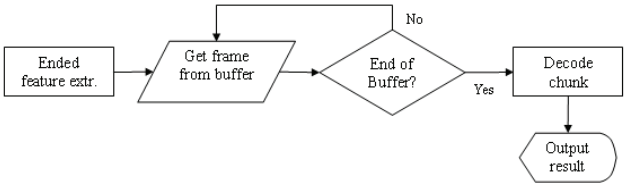


Fig. 10. Flow diagram of modified HVite process

C. Live Recognition Module

In the recognition phase, feature frames extracted in the previous steps are sent to the MSHMM decoder to decide which word is spoken. For this step, HTK library is used. The general idea in this step is to modify recognition application (HVite) of HTK toolkit to support live recognition with multi-stream user data and embed it into the application. Normally HVite supports live recognition with audio data when started without an input file and outputs recognition results to standard output. During live recognition HVite checks whether the input is from audio hardware and works only if so. The flow diagram of the regular HVite live audio recognition is in Figure 9.

Soshi Iba from CMU [21] presents some modifications to HVite to support live recognition from standard console input by bypassing the audio-only check and feeding the recognition stream with characters entered from the keyboard. We have taken those modifications further so that recognition stream is fed from the features extracted and stored on the buffer. Also the output is passed to the main program, so that the program handles it and performs necessary commands.

Normally for live recognition HVite performs a noise detection procedure and extracts chunks of spoken speech. It then performs necessary feature extractions on grabbed chunk and sends this chunk of features to the recognizer. Since we are already capturing only a chunk of speech and extracting features, the crucial idea is to fork HVite's recognition procedure; and feed the extracted features on the buffer instead.

For every frame that is processed, features of the following frame are used to fill an array, starting from array element 1, that is taken as input by HVite. The only necessity that was not foreseen on this procedure was to fill the elements of the array that divide streams with zero. For researchers interested in similar work, we recommend to take care in this situation.

The flow diagram of the modified HVite MSHMM live recognition procedure is in Figure 10.

VI. Running The Application

As mentioned, current state of application performs audio-visual recognition with audio (MFCC) and visual (mouth ROI DCT) features. The initial screenshot of the application can be seen in Figure 11. When initialized, it waits for the user to click the "Begin" button to begin capturing. Also, the user can prefer to do recognition or

only feature file extraction by checking the check box labeled "Just Save".

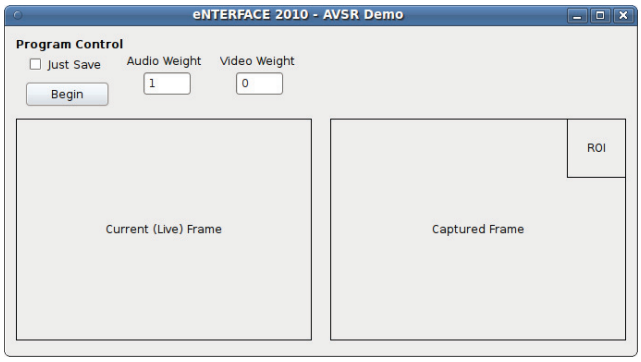


Fig. 11. Initial screenshot of the application

In either condition, after the button is clicked capturing process is triggered. During capturing, captured visual frames can be seen on the application as seen in Figure 12.

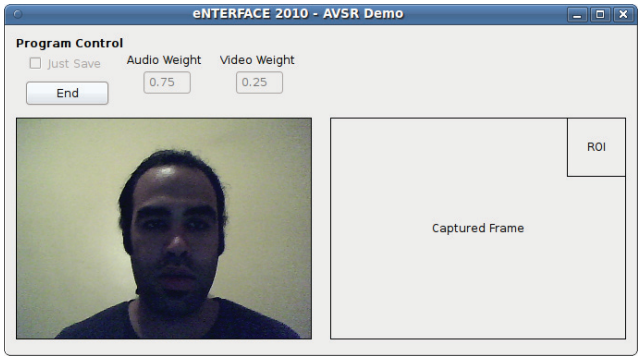


Fig. 12. Screenshot of the application while capturing data

After capturing ends, audio and visual frame synchronization is performed as proposed in Section III-G and if user had preferred extracting feature files only two different HTK format feature files are saved on user's computer. If recognition is performed, recognized command is run on the computer. The result of a recognition process can be seen in Figure 13. For recognition, user can change stream weights for audio and visual data with the software using corresponding text boxes.

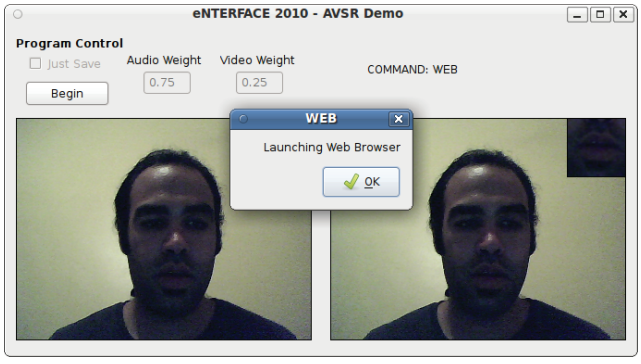


Fig. 13. Screenshot of the application when recognition ends

The screenshots are taken while the application is running live on a Toshiba laptop, running Ubuntu Linux 10.04, which is a fine example of the application's and related libraries' multi-platform support.

VII. Experimental Results

To get numerical results and see whether visual features help recognition, we have collected small number of data and performed offline experiments. The online recognition result can be seen on previous section. We have collected a dataset of ten spoken English words and the offline recognition experiment is performed for this ten words. The dataset is collected with a Toshiba laptop, with a moderate integrated web camera and a built-in microphone which puts a hiss noise to the recorded audio. The files in HTK feature file format are collected with the application, where in each file the user says all words to be recognised. A subset of the recorded files are used as training set and a hidden Markov model is trained on this set. To obtain an audio-visual MSHMM, we concatenate visual data to audio data and train the multi-stream model using single-pass retraining [8] from the audio-only HMM. Then the remaining files are used as the test set and recognition performance of the trained model is evaluated on this set of data. The results for the experiment are on following Table II.

TABLE II
Offline recognition results

Audio Weight	Video Weight	Accuracy
1.00	0.00	80.00
0.875	0.125	90.77
0.75	0.25	86.15
0.625	0.375	76.92
0.50	0.50	66.92
0.375	0.625	53.85
0.25	0.75	51.54
0.125	0.875	53.85
0.00	1.00	36.92

The results may be interpreted such that; as seen on Table II, if the recording environment has implicit noise (in this case due to the hiss in microphone) using visual features improves accuracy. This shows the advantage of fusing audio and visual features and positive effect on the recognition accuracy.

It should be reminded that these tests are performed with limited number of training data. A higher and more diverse data set where recordings are done with different people, at different places and times, a deeper analysis can be made. However visual features are promising for recognition when audio data is insufficient.

VIII. Conclusions and Future Work

We have developed a software during the eNTERFACE'10 workshop that demonstrates how to control a computer with audio visual speech recognition. Speech recognition phase is performed with both audio and visual channels, and during recognition user can choose which channel is emphasized with what weight. Also user can use the software to extract source files to train new models. The application uses multi-platform, free and open source libraries and development environment. This allows other researchers who wish to benefit from the project at maximum. We also propose using tandem approach and give results done with offline data. The results for tandem streams are also promising.

The source file for the entire project, presentations and a demo video can be found on eNTERFACE 10 web site [22] and project home page [23]. One exception for the source files is that, live recognition module which is modified from HTK toolkit cannot be distributed.

For future work, automatic detection of SNR and speech activity may improve usability and continuous recognition such that; when

activity is detected recognition begins and ends when the activity ends. Also detecting audio SNR and quality of visual information can help to determine stream weights automatically. Also coupled hidden Markov models [24] may be employed to overcome state asynchrony in audio and visual channels and improve accuracy.

IX. Acknowledgements

This research is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under the scientific and technological research support program (code 1001), project number 107E015 entitled "Novel Approaches in Audio Visual Speech Recognition".

Alexey Tarasov is funded by Science Foundation Ireland under Grant No. 09-RFP-CMS253.

References

[1] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models", *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, Jan 1986.

[2] R. A. Gopinath, "Maximum likelihood modeling with gaussian distributions for classification", in *Proceedings of ICASSP*, 1998, pp. 661–664.

[3] M. J. Tomlinson, M. J. Russell, and N. M. Brooke, "Integrating audio and visual information to provide highly robust speech recognition", in *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*, Washington, DC, USA, 1996, pp. 821–824, IEEE Computer Society.

[4] Hynek Hermansky, Daniel P. W. Ellis, and Sangita Sharma, "Tandem connectionist feature extraction for conventional hmm systems".

[5] Christopher J.C. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[6] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1999.

[7] Stéphane Dupont and Juergen Luettin, "Audio-visual speech modeling for continuous speech recognition", *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 141–151, 2000.

[8] Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*, Cambridge University Press, 2006.

[9] "Qt - cross platform application and ui framework", `http://qt.nokia.com`.

[10] W. Burger and M.J. Burge, *Undergraduate Topics in Computer Science. Principles of Digital Image Processing. Core Algorithms*, Springer-Verlag London, 2009.

[11] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental", *Pattern Recognition and Artificial Intelligence*, pp. 374–388, 1976.

[12] "Portaudio - portable cross-platform audio api", `http://www.portaudio.com`.

[13] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck, *Discrete-time signal processing (2nd ed.)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1999.

[14] S. Steidl, *Automatic Classification of Emotion-Related User States in Spontaneous Children's Speech*, Phd, Universitat Erlangen-Nurnberg, 2009.

[15] "Libxtract library", `http://sourceforge.net/projects/libxtract/`.

[16] "Opencv - open source computer vision library", `http://opencv.willowgarage.com`.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–511–I–518 vol.1, 2001.

[18] Erdogan H. Topkaya I.S., "Using multiple tandem streams in audio-visual speech recognition (submitted)", in *ICMLA '10: Proceedings of the International Conference on Machine Learning and Applications, 2010 IEEE International Conference*, Washington, DC, USA.

[19] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001.

[20] J. Wagner, Vogt T., and Andre E., "A systematic comparison of different hmm designs for emotion recognition from acted and spontaneous speech", 2007, vol. 4722 of *Lecture Notes in Computer Science*, pp. 114–125, Springer.

[21] "Htk3.0 batch and streaming recognition demo", `http://www.cs.cmu.edu/~iba/research/htk3/index-e.html`.

[22] "enterface'10 summer workshop in amsterdam", `http://enterface10.science.uva.nl/`.

[23] "Novel approaches in avsr project at vpa", `http://vpa.sabanciuniv.edu/phpBB2/vpa_views.php?s=4&serial=49`.

[24] Ara V. Nefian, Luhong Liang, Xiaobo Pi, Xiaoxing Liu, and Kevin Murphy, "Dynamic bayesian networks for audio-visual speech recognition", *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 1, pp. 1274–1288, 2002.

I. Saygin Topkaya is a Ph.D. student and a research assistant in Electronics Engineering department at Sabanci University. He received his B.S. and M.S. degrees from Mathematical Engineering at Yildiz Technical University at 2004 and 2008 respectively. His research interests are computer vision, audio-visual speech recognition and face recognition. Currently he is involved in the project "Novel Approaches in Audio-Visual Recognition" and continues his Ph.D. study under the supervision of Asst. Prof. Hakan Erdogan.

M. Berkay Yilmaz is a Ph.D. student in Computer Science and Engineering department at Sabanci University. He received his B.S. degree from Computer Engineering at Bahcesehir University and his M.S. degree from Mechatronics Engineering at Sabanci University. Currently he is involved in the project "Novel Approaches in Audio-Visual Recognition".

Umut Sen is an MS student at Electronics Engineering of Sabanci University and also a member of VPA. He received my BS degree from also electronics engineering of SU. Currently he is involved in the project "Novel approaches in audio-visual speech recognition" under supervision of Asst. Prof Hakan Erdogan. He is also interested in classifier ensembles, feature selection and transformation, speaker verification.

Alexey Tarasov was born in Riga, Latvia in 1984. He has received his B.Sc. and M.Sc. degrees in Computer Science from Transport and Telecommunication Institute. His Master thesis was devoted to the diagnostics of aircraft engines, but now he works on his Ph.D. in the field of automatic emotion recognition from speech in Dublin Institute of Technology with Dr Sarah Jane Delany and Dr Charlie Cullen.

Hakan Erdogan is an assistant professor at Sabanci University in Istanbul, Turkey. He received his B.S. degree in Electrical Engineering and Mathematics in 1993 from METU, Ankara and his M.S. and Ph.D. degrees in Electrical Engineering: Systems from the University of Michigan, Ann Arbor in 1995 and 1999 respectively. His Ph.D. was on developing algorithms to speed up statistical image reconstruction methods for PET transmission and emission scans. He was with the Human Language Technologies group at IBM T.J. Watson Research Center, NY between 1999 and 2002 where he worked on various internally funded and DARPA funded projects. At IBM, he focused on the following problems of speech recognition: acoustic modeling, language modeling and speech translation. He has been with Sabanci University since 2002. His research interests are in developing and applying probabilistic methods and algorithms for multimedia information extraction. Specifically, he is interested in sequence labeling, speech recognition and developing algorithms for learning.

An Affect-Responsive Interactive Photo Frame

Hamdi Dibeklioğlu, Ilkka Kosunen, Marcos Ortega Hortas, Albert Ali Salah, Petr Zuzánek

***Abstract*— We develop an interactive photo-frame system in which a series of videos of a single person are automatically segmented and a response logic is derived to interact with the user in real-time. The system is composed of five modules. The first module analyzes the uploaded videos and prepares segments for interactive play, in an offline manner. The second module uses multi-modal input (activity levels, facial expression, etc.) to generate a user state. These states are used by the internal frame logic, the third module, to select segments from the offline-generated segment dictionary, and they determine the response of the system. A continuous video stream is synthesized from the prepared segments in accordance with the modeled state of the user. The system logic includes online/offline adaptation, which is based on stored input-output pairs during real-time operation, and offline learning to improve the system response. The fourth module is the application interface, which deals with handling the input and output streams. Finally, a dual-frame module is described to enhance the use of the system.**

I. Introduction

In this paper we describe a dynamic responsive photo frame. This system replaces a traditional static photograph with a video-based frame, where short segments of the recorded person are shown continuously, depending on the input received from the sensors attached to the interactive frame.

The prototypical scenario we consider is the photograph of a baby, set up in a different location, for instance in the living room of the grandparents. While there is no one around, the baby is asleep in the photo frame. Once a viewer arrives, the baby 'wakes up', and responds to the multimodal input received from its viewer. To realize such a system, we propose methods to automatically analyse and segment a number of video sequences to create a response dictionary, combined with a real-time affect- and activity-based analysis tool to select appropriate responses to the user. We then propose a number of system extensions and describe an evaluation methodology.

This system has a number of precursors. A responsive interactive system was proposed in [1], called an Audiovisual Sensitive Artificial Listener. It is a system in which virtual characters react to real users. Facial images and voice information in the videos are used to extract features, which are then submitted to analysers and interpreters that understand the user's state and determine the response of the virtual character. Hidden Markov Models (HMMs) are used in sequential recognition and synthesis problems. In [2], a dialogue model is proposed that is able recognize the user's emotional state, as well as decide on related acts. A Partially Observable Markov Decision Process approach is used with observed user's emotional states and actions.

A project which brought some interaction to photographs is the Spotlight project of Orit Zuckerman and Sajid Sadi, developed at MIT MediaLab[1]. In this project, 16 portraits are placed in a 4×4 layout. Each portrait has nine directional temporal gestures (i.e. one of nine images of the same person can be displayed in the portrait at any given time), which give the appearance of looking at one of the other portraits, or to the interacting user. The user of the system can select a portrait, at which point the remaining portraits will 'look' at it. This project demonstrates the concept of an interactive photograph with static content. While the combination of portraits create novel patterns all the time, the language of interaction is simple and crisply defined.

Another interactive photo frame project is the "Portrait of Cati" by Stefan Agamanolis, where the portrait in question can sense the proximity of the spectator, and act accordingly [3]. When no one is close to the portrait, Cati displays a neutral face. When someone approaches, it selects a random emotion, and displays it in proportion to the proximity of the spectator. If the selected expression is a smile, for instance, the closer the spectator comes, the wider Cati will smile. A similar project is the Morface installation, where an image of Mona Lisa was animated based on the proximity of the interacting person [4]. In this project camera-based tracking is used to determine proximity and head orientation of the user.

The system described in here is different in several aspects from the systems discussed in the literature. In our model the responses of the system are not fixed, but grow in time as the user uploads new videos. In this manner, the system maintains novelty. The two interactive systems we just described are suitable for art installations, but we target a home application, for which novelty plays an important role. Another aspect is that we use real videos in the systems output, with no manual annotation. This is much more challenging than producing appropriate responses through a carefully engineered synthesis framework, where the system has control over the output.

The bottleneck in our system is the real-time interaction, therefore we need to work with lightweight features. We first inspect simple and easy-to-recognize signals, and move to recognition of more complex stimuli. The second aspect that makes our work novel is that the response of the system is not manually (and precisely) defined. A fully automatic segmentation procedure is proposed to create self-contained response patterns, for which the precise semantics is not known at the onset. Our goal is to create a consistent system, in which certain user behaviour is used to produce a certain system response in a consistent manner, and the user is the primary driver of the interaction semantics.

The primary modality we use for real-time analysis is the facial expression of the user. At the core of our real-time module is the eMotion system, which recognizes six basic emotional expressions in realtime [5], [6]. This system uses a Bézier volume-based tracker for the face and a naïve Bayes classifier for expression classification. In a similar vein, Kaliouby et al. previously proposed a MindReader API which models head and facial movements over time by Dynamic Bayesian Networks, to infer a person's affective-cognitive states in real time [7]. In [8] a real-time emotion analysis system was proposed that used an efficient facial feature detection library in conjunction with a number of physiological signals. In the last few years, facial expression and action recognition have seen great improvements. For additional information on facial expression recognition, see [9], [10], [11].

This report is structured as follows. In Section II we describe the proposed system, its separate modules, and its use-cases. Section III describes the algorithmic aspects for each of the modules of the system. Section IV describes the experimental methodology and the assessment of the proposed algorithms within the application context. As the complete system implementation was not completed until

H. Dibeklioğlu and A.A. Salah are with the Informatics Institute, University of Amsterdam, the Netherlands. I. Kosunen is with Helsinki Institute of Technology, Finland. M. Ortega is with University of A Coruña, Spain. P. Zuzánek is with Czech Technical University, Czech Republic.

[1]http://ambient.media.mit.edu/people/sajid/past/spotlight.html

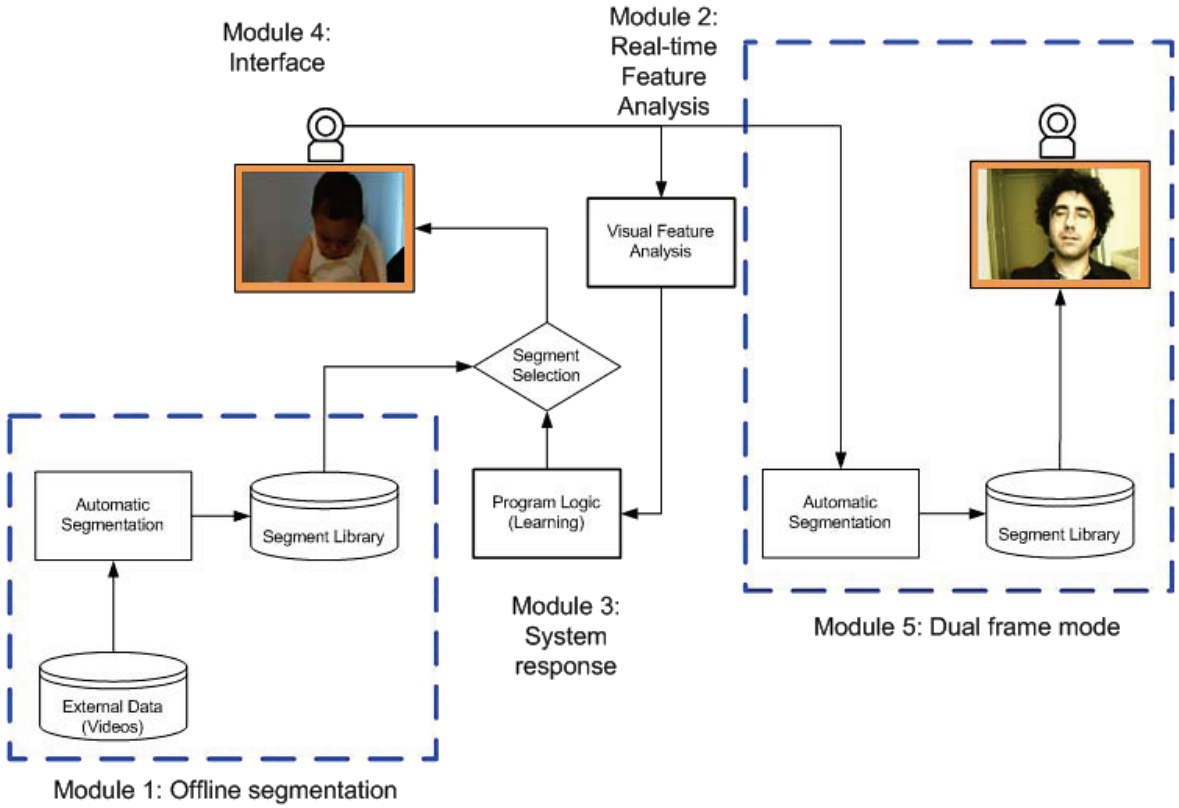


Fig. 1. The overview of the operation of the system. In the dual-frame mode, each frame is used to record new videos that are automatically segmented and added to the segment library of the other frame, establishing an asynchronous communication channel.

the end of the eNTERFACE Workshop, a usability study was not performed. However, such a study was planned during the Workshop. Finally, we conclude in Section V and summarize possible future directions.

II. Design of the Affect-responsive Photo Frame

In this section we describe the logic and the design choices for the affect-responsive photo frame, and introduce the separate modules. Our purpose is to create an emotional/personal digital artifact for continued use. This artifact is designed to adapt to each of its users, as well as prompt the user to adapt to its behavior by guiding the user. We will describe the whole system through the prototypical baby-grandmother scenario. This will help distinguishing the two analysis modules that are similar in principle, but work in offline and online modes, respectively.

A. Overview

The first part of the system is the offline segmentation module. The purpose of this module is to create a response segment library, composed of short video fragments. The input is any number of uploaded videos. In the prototypical use-case, these are the videos of the baby. These videos are analyzed in an offline fashion, and the segments are stored in a segment library. During interaction, the system will play these segments in a particular order.

The second module is the affect and activity analysis module. Here the visual (and in the future audio) input from the user is analyzed in real-time, and a feature vector is generated. This is the module that processes the behaviour of the grandmother in the use-case.

The feature vector is used by the third module, which is the system response logic. The features computed in the second module are used to select an appropriate video segment from the segment library. This module also incorporates learning, to fine-tune its response over time. The system uses its offline period to execute an unsupervised learning routine for this purpose.

The fourth module is the interface. The segments are displayed to the user in the photo frame, depending on the user input. For instance, a smile will trigger a response from the frame, but since we have no mechanism to interrupt the response of the system as soon as new input arrives, a faster feedback mechanism is integrated to the frame in form of coloured glyphs, displayed under the image. Each system response is associated with one glyph, and the brightness of the glyph indicates the proximity of users behaviour to the activating input for that particular response. Thus, if a response is triggered by a smile, a wide smile will activate its glyph immediately, and the response will be played once the current sequence ends playing.

Finally, the fifth module is implementation of the dual frame mode. Here there are two frames, in different locations. Each frame records new segments when it is interacting with a user, analyses those segments, and sends them over the Internet to the other frame system. These segments are added to the segment library of the other frame. They also come with some ground truth, we already know what kind of input elicited these responses in the first place, so we can associate their activation with similar input patterns. This design takes care of content management, and provides constant novelty to the system. Figure 1 shows the overall design of the system, complete with the dual-frame mode.

B. Offline Segmentation Module

The task of the offline segmentation module is to automatically generate meaningful and short segments from collected videos. These

are stored with indicators of affective content and activity levels. Segmentation errors here are not of great importance, as the synthesis module will eventually use all footage material.

The segmentation module uses optical flow calculations to find calm and active moments in the video. Each active moment of a specific length, surrounded by calm moments, is considered as one event and labeled as an active segment. Also, each calm period of sufficient length is labeled as a calm segment. Due to the generic nature of the optical flow calculation, the module is able to detect not only changes in facial expressions, but also events such as hand gestures (waving to the camera) and head movements.

Since we cannot assume a neutral initial pose, or an occlusion-free face area for the duration of the video, assessing facial expressions in these uncontrolled segments is really difficult. Furthermore, our use-case involves a baby, for which the expression analysis requires special training due to different facial morphology. Our experimental results have shown that the optical flow based segmentation creates segments similar to manual segmentation.

C. Real-time Feature Analysis Module

The real-time analysis module is motivated by the need of the system to analyze and characterize user behaviour in order to provide an appropriate response to any particular behaviour. Keeping this in mind, this module can be considered as the data source of the system, as it receives signals from the user and processes them to determine affect- and activity-based content. Since the data are gathered during real-time operation, the module must be able to analyze and process data in a real-time fashion, within reasonable computational assumptions.

The feature analysis module combines the input data into a single feature vector aimed to characterize the current action taken by the user of the system. Modelling the action as a feature vector has the critical advantage that it allows to generate an action space covering the possible feature vector values. This space can be further used to improve system responses to a specific user using machine learning techniques.

In our initial design of the system, we have focused on the following aspects of the user behaviour to be able to model a significant and complete set of different actions:

- Face: The location of the face is the first and most important feature of the system. It allows us to detect the presence of a user to initiate a session, and at the same time it offers information during the session such as movement with respect to camera's frame of reference, and proximity of the user.
- Eyes: The location of the eyes gives us information about the gaze direction of the user. In a system with synthesized responses, matching gaze direction with the user (shared focus of attention) or following the user's location with the gaze are both important for realistic interaction. Since we do not assume any control over the stored segments, there is no meaningful way we can match the gaze information with appropriate segments. However, we do know where the strongest action takes place in each segment, and the gaze information can potentially be matched to such a cue.
- Motion: The activity level of the user is a lightweight feature that can be usefully employed to characterize actions. We divide the face area into a grid and measure the amount of activity in each cell of the grid. This gives us a granular indication of facial activity levels.
- Expression: Facial expression analysis is computationally costly. In our prototype we detect the six prototypical facial expressions (joy, sadness, anger, fear, surprise, disgust). Our system gives soft membership values for each category (including neutral) at 15 fps.

In future, we plan to take more input channels into consideration, like color information (in order to detect presence of some predominant color in the scene, possibly indicating an object) or audio cues from the user.

1) Feature vector components: Fig. 2 shows the information that the real-time analysis module extracts from the input data in a given frame $\mathcal{F}$ in order to construct the feature vector. We also need to consider the action of the user in some interval of time to model the evolution of activity and movement. Therefore, we consider a past frame $\mathcal{F}'$, typically two or three frames prior to $\mathcal{F}$. The feature vector computed at $\mathcal{F}'$ is used in conjunction with the feature vector computed at $\mathcal{F}$ to determine the system response. In addition to these location and activity based features, we use facial expression analysis to provide us with the amount of expression present in each frame. This additional information comes as a normalized vector containing the amount for each one of the six basic facial expressions, plus the neutral expression (represented as $E_1 \dots E_7$). Table I summarizes the feature vector components related to the computations from the frame data.

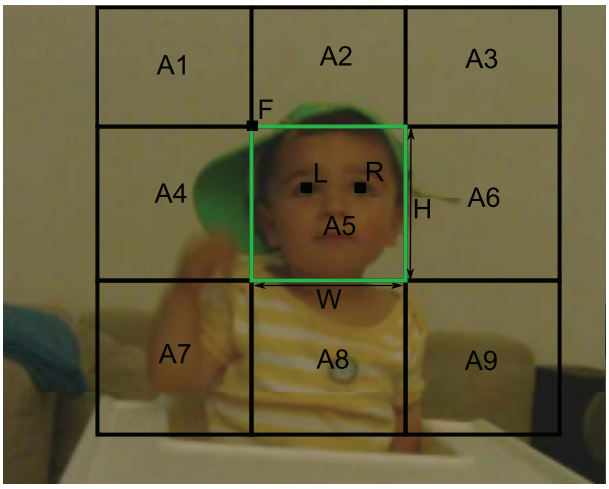


Fig. 2. Features gathered from the visual input of the user for a particular frame. (F_x, F_y) represents the coordinates of the left corner of the face region with respect to the image borders, W is its width and H is its height. $L = (L_x, L_y)$ and $R = (R_x, R_y)$ represent the left and right eye locations, respectively. $A_1 \dots A_9$ quantify the motion activity in each of the nine regions around the face. These nine parameters are measured as vectors, the magnitude ($|A|$) being the amount of average activity in the region and the direction (A_ρ being the mean direction for each region).

D. System Response Module

The system response determines the quality of interaction. If the automatic segmentation is successful, we have a number of short segments that can be played in any sequence. This forms a baseline for the operation of the system. The purpose of the system response module is to improve on this baseline by evaluating the user input in real-time, and by producing consistent and meaningful responses.

We have selected a finite state machine as the abstract representation of the system's operation in this module. This is the simplest possible model for interaction, where input and output relations are clearly (but probabilistically) indicated.

1) Simple prototype: As a simple first prototype, we have developed a simple, two-state finite state machine. The transition between the states were made to depend on the results of the Viola-Jones face detector (i.e. the input consisted of a Boolean variable representing

TABLE I

DESCRIPTION OF THE 41 FEATURES USED TO BUILD THE FEATURE VECTOR FOR FRAME $\mathcal{F}$. FEATURES ARE DEFINED IN TERMS OF THE COMPUTATIONS OF FRAME $\mathcal{F}$ AND REFERENCE FRAME $\mathcal{F}'$. THESE COMPUTATIONS CORRESPOND TO THE ONES EXPRESSED IN FIG. 2.

Index	Calculation	Definition
F_1, F_2	F_x, F_y	Face region left corner coordinates
F_3, F_4	W, H	Width and height of face region
F_5, F_6	$C_x - C'_x$, $C_y - C'_y$	Translation of the face region from $\mathcal{F}'$ to $\mathcal{F}$
F_7, F_8	$\frac{W}{W'}$, $\frac{H}{H'}$	Scale factor of the face region from $\mathcal{F}'$ to $\mathcal{F}$
F_9, F_{10}	L_x, L_y	Left eye center coordinates
F_{11}, F_{12}	$L_x - L'_x$, $L_y - L'_y$	Left eye center translation from $\mathcal{F}'$ to $\mathcal{F}$
F_{13}, F_{14}	R_x, R_y	Right eye center coordinates
F_{15}, F_{16}	$R_x - R'_x$, $R_y - R'_y$	Right eye center translation from $\mathcal{F}'$ to $\mathcal{F}$
$F_{17} \dots F_{25}$	$\lvert A_1 \rvert \dots \lvert A_9 \rvert$	Magnitude of motion vectors in regions A_1 to A_9
$F_{26} \dots F_{34}$	$A_{1\rho} \dots A_{9\rho}$	Motion vector directions for regions A_1 to A_9
$F_{35} \dots F_{41}$	$E_1 \dots E_7$	Amount of basic expressions present in the current frame

"face detected' and 'face not detected'). Fig. 3 depicts this two-state machine. We have used two expressive face action sequences ('Sad' and 'Smile', respectively) from the Cohn-Kanade database [12]. The advantages of using these sequences are that they are normalized with respect to face location and size, well-illuminated, and the expressions start from a neutral face and evolve into the full manifestation of the expression.

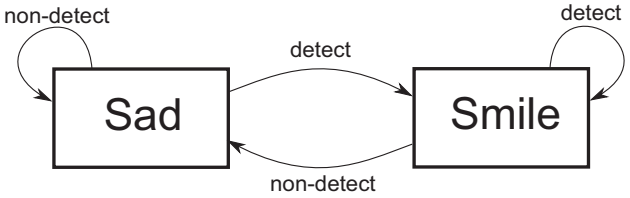


Fig. 3. Scheme of the two-state machine that changes the response of the system according to the results of the Viola-Jones face detector.

This prototype helped us to inspect the behaviour of the system under very simple operating principles, and led to the following observations:

- **Neutral state:** It is unnatural to repeat a video segment multiple times, as the jump from the last frame to the first frame induces an abrupt motion. In the prototype, we ensured the smooth transition between segments by playing them forwards and backwards in a single output cycle. Thus, any transition from the 'Sad' state to the 'Smile' state occurred when the face was displaying a neutral expression. We have decided to use such a 'neutral state' in all our state transitions. We define the neutral state as a frame with very low activity, so that the switch from a forward play to the backward play of the segment has minimum unnatural motion. We have also experimented with morphing between segments to have a smooth transition, but it is difficult to ensure a proper registration of anchor points between frames automatically to have a natural and smooth morphing sequence.
- **Uninterrupted play:** While the response logic requires the system to change behaviour as soon as a new user input is registered, it is unnatural to interrupt a sequence in progress and switch to another sequence. We decided to switch the segments (make a state transition) after the current segment is played completely. For the acknowledgement of the user input, a supplementary indicator is designed. This will be described shortly.
- **System responsivity:** With uninterrupted play, there is a related issue of the length of the video segments. Longer video segments means that the system response is delayed, while the segment is run to its end. A solution might be to eliminate longer sequences from the segment library, or to make them rare events in the operation of the system.
- **Video transitions:** When we have a transition between segments that naturally follow each other, the state transition is very smooth, as expected. However, switching to a distant segment of the same video session, and even more prominently, switching to a segment of another video session can be sharp and unnatural. These transition artifacts should be eliminated using a smoothing or blur function during the transitions. In [13] a subspace method is proposed to control real-time motion of an object or a person in a video sequence. The low-dimensional manifold where the images are projected can be used to define a trajectory, which is then back-projected to the original image space for a smooth transition. While this method is promising for controlling transitions between segments, the subspace projection will not be very successful with the dynamic and changing backgrounds we deal with. Subsequently, we use a much simpler scheme. If we have a transition from frame $\mathcal{F}_1$ to frame $\mathcal{F}_2$, we use an exponential forgetting function to synthesize transition frames, given by equation:

$$\mathcal{F}_3 = \alpha \cdot \mathcal{F}_1 + (1 - \alpha) \cdot \mathcal{F}_2 \tag{1}$$

where $\alpha \in (0; 1)$.

2) Design of the finite state machine: According to the observations we made following the prototype experiment, we have developed a more extensive finite state machine for the affect-responsive photo frame, depicted in Fig. 4.

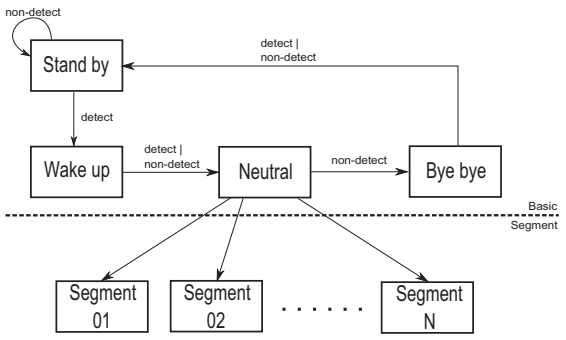


Fig. 4. Scheme of the finite state machine for system response. The two kinds of states are distinguished by the dotted line separator.

In this current implementation, we distinguish between *Basic states* and *Segment states*, respectively.

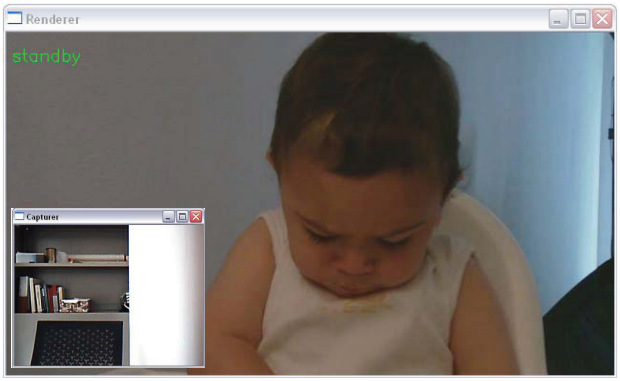
- **Basic states** are used to provide a general and consistent outlook to the system. When the system is not in use, a default state of low activity is played in loop. In the prototypical use-case, this state would depict the baby asleep in the frame. When a person is present, the baby wakes up, and normal operation is resumed. When the interacting user is absent for a long period, the system returns to the sleep mode. The basic states make sure that this skeleton response is properly displayed. They are assigned manually, although their segmentation need not be manual. The transition from one basic state to another basic state depends solely on the input from the face detector.
- **Segment states** constitute the dynamic part of the finite state machine. Each segment state S_i is associated with one video segment V_i from the segment library, as well as an expected feature vector F_i that will guide the activation of the segment. In the first working prototype of the system, implemented during the eNTERFACE, we have assigned the expected feature vectors randomly, by setting the activity and location based values to zero and setting one or two of the facial expression dimensions to larger values. Thus, basic expressions were used to elicit responses from the system. The segment V_i is activated when the feature vector describing the user's activity is close to the expected feature vector F_i. The 'closeness' here is described statistically, by specifying a Gaussian distribution around each expected feature vector, and admitting activation if the feature vector computed from the user's activity is close to the mean by one standard deviation.

To better understand and remember the user's response for each segment, a game-like strategy is used, where the responses of the system are 'unlocked' one by one. This means that the user has to discover the correct response expected by the system for each new video segment that is shown on the frame. At the beginning, all segments are locked. Once the correct response for the segment in line is found, the particular segment becomes active, and it can always be re-activated by producing the same response.

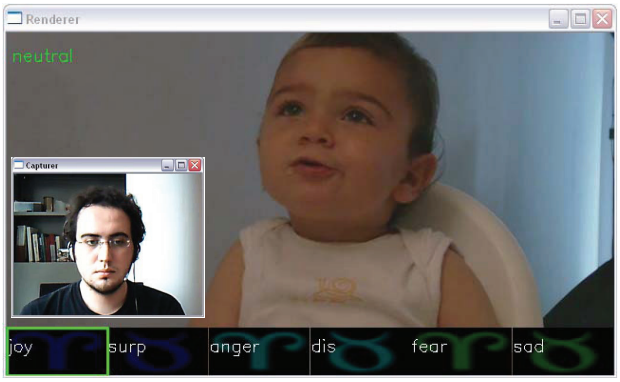
E. Interface

The interface of the affect-responsive photo frame contains a feedback mechanism to allow the user to see the immediate effect of its actions. This was necessary, as we treat each video segment as an integral entity, and play them in their entirety. Longer segments reduce the responsiveness of the system. In this section we describe our solution based on coloured glyphs, displayed under the photo frame. We also describe the external software packages we made use of to run the system on a stand-alone computer system.

1) Glyphs responses: The developed system aims to provide two-sided adaptation between the user and the machine. Two-sided adaptation means that it is not only the machine that learns the patterns of the user, but also that the user learns the reactions of the machine for different patterns. For this purpose, our system shows glyph responses in real time for patterns derived from the actions of the user at a specific moment. Each segment is pre-assigned to a glyph, which shows the relation between the user's behaviour and the system response, which is encoded by the intensity of the glyph. Higher intensity means that the user activity comes close to the activity that is associated with the particular segment. When the intensity reaches its maximum, the segment is activated. It is shown to the user once, and the user response that elicited the activation of the segment can be repeated for re-activation of the segment at later times.



(a)



(b)

Fig. 5. (a) The stand-by mode and (b) the interaction mode of the system. The lower left corner shows the current camera input to the photo frame as a diagnostic tool.

Fig. 6 shows the system with the glyph responses for each segment. The order of the glyphs (from left to right) reflects the order of segments in the unlock queue. The third glyph glows bright in the example, which means that the current activity of the user is very close to the activity pattern that activated the third segment. The green bounding box around the fourth glyph shows that this is the next segment to be activated, and if the user wants to unlock this segment, he or she should watch this glyph for intensity changes, and adjust its behaviour to increase this intensity. The glyphs on the left side of the green bounding box are already unlocked, and at any given moment, the user can elicit these responses from the system by the same behaviour that was used to unlock the segment initially. Responses for segments to the right of the green bounding box are not known to the user yet.

2) External software: To enable facial expression analysis in our system, we have used the approach proposed in [6]. There is an existing software implementation of this method, packaged into the commercial eMotion application[2]. This program analyses a face image, and classifies the facial expression into basic emotional categories. We have modified some output channels of this expression analysis system and prepared a separate executable to avoid running face detection twice. In the prototype we have prepared, the modified eMotion software runs in addition to the main program, and feeds the facial analysis results to our system over a Telnet connection. Since both interaction and eMotion systems need camera input, we have used a third party camera splitter driver (SplitCam software) which clones the camera input for both applications.

[2]http://www.visual-recognition.nl/

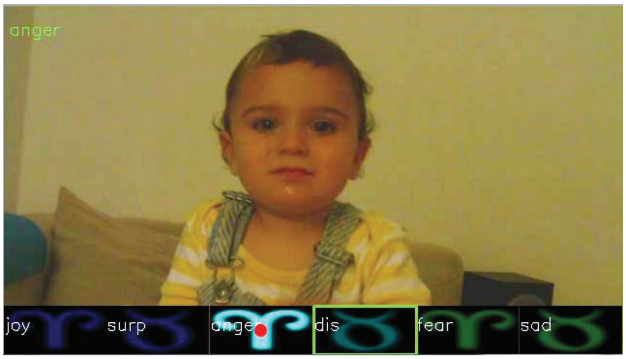


Fig. 6. The output window displays the segment and the glyphs below it. The red circle in the center of the third glyph shows the currently playing segment. In the top-left corner, the name of the currently playing segment is displayed. This is a prototype where each segment is named with basic expression categories. This information is normally not available to the system, as the segmentation is automatic.

F. Dual Frame Mode

The principle behind the dual-frame mode resembles that of the PhotoMirror appliance [14]. In PhotoMirror, a camera is hidden behind a mirror in a home setting, which can record segments of the inhabitants life, and play them back on the surface of the mirror (or another mirror). Similarly, the dual-frame mode of our system implies an asynchronous communication between two persons.

Consider our example scenario with the baby and the grandmother, and add to it a time-differential, where the baby lives in another continent. While the grandmother uses the interactive photo frame in her house, the system will record short segments of her activity (where the face detector is active) and create a segmented behavior library for the grandmother. These segments will be played on a second frame, placed in the baby's room. Through this symmetrical setup, we will also have a kind of action-response ground truth; the segments recorded from the grandmother's frame will be associated to particular segments of the baby. Then, these associations can be used to weakly guide the response patterns. Furthermore, each usage of the frame will send a sequence of new segments to the other frame, taking care of automatic content update for improved novelty.

III. Algorithmic Aspects

A. Offline Segmentation Module

The optical flow algorithm can be controlled in various ways depending on the type of segmentation that is desired. First there is the question of whether the optical flow should be calculated between two consequent frames, or a longer period, which might be necessary if the video footage is very static. Secondly, the number of tracked features can be adjusted: in videos with lots of small, uninteresting motion, the algorithm could be set to track only the most important features. Furthermore, the distance between two unique features can be scaled, and the maximum effect of a given feature can thereby be made greater or smaller. This provides robustness against outliers, so that a single large deviation in a given feature, which may be the result of an outlier or noise, does not overly affect the result. With all these options, the module can be used to segment a wide variety of video content. We now discuss several aspects of this module.

1) Optical flow and motion energy: The optical flow calculations were performed with standard routines of the OpenCV library[3]. Optical flow is calculated by selecting the number of points or features in one frame image, and tracking the distance these points have moved in another frame. The tracked features can be selected in a variety of ways [15], but we used the Shi-Tomasi corner detection algorithm [16]. Once the features are selected, they are used using the Lucas-Kanade method for optical flow estimation [17]. The resulting optical flow for each feature in the frame (up to a pre-specified number of features) is then summed to produce a total amount of optical flow for each frame. Because we are interested in events that last for several seconds, the optical flow data are then smoothed using a moving average window to get rid of noise, as well as large fluctuations. This procedure is illustrated in Fig. 7.

After the smoothed optical flow data are generated, the algorithm goes through the data and finds extended periods of activity and calm, and generates both calm and active segments based on this information. The main problem is automatically selecting reasonable thresholds for what is considered an activity and what is not. This is done by defining the average activity as the amount of total optical flow in a frame, and then by taking a certain percentage of this amount to be the threshold for activity segmentation. This allows the algorithm to work with both very active videos, as well as comparatively static ones.

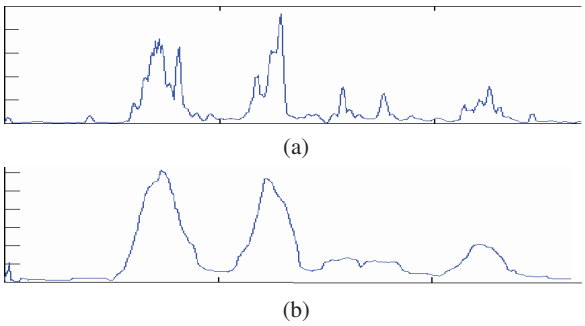


Fig. 7. (a) The original optical flow curve and (b) its smoothed version.

2) Frontal face detection: Apart from activity analysis, we rely on face information for both offline and online processing. The first step for this purpose is face detection. While it is possible to do a pass over the video segments that are processed offline to find the best (frontal) face, and combine this with tracking to provide robust face localization, this approach was not implemented. The ideal combination of frame-by-frame face detection and tracking is a possible extension left for the future work.

Because of its proven reliability, we have selected the well-known Viola & Jones algorithm for face detection [18]. For better accuracy we have used the improved version of Viola & Jones algorithm as proposed by Lienthart and Maydt [19]. In this improved version, 45° rotated Haar-like features (see Fig. 8) are used in addition to the original set of Haar features, and a post optimization of boosted classifiers is performed. While rotated Haar-like features increase the discrimination power of the framework, post optimization of the boosted classifiers provides for reduced false alarms.

The Viola & Jones method can be used to detect rotated faces with a cascade trained for this purpose. In general, frontal face images are easier to analyse, and the expression analysis module used in this study needs a frontal face at the initialization step. Therefore, we have used only frontal face cascades to recognize nearly frontal faces.

B. Real-time Feature Analysis Module

In this section we describe the techniques employed in the real-time feature analysis module in order to compute the feature vectors

[3]See David Staven's excellent tutorial "The OpenCV Library: Computing Optical Flow" for more information.

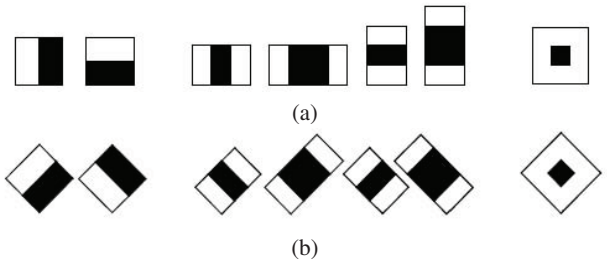


Fig. 8. (a) Haar-like edge, line, and center-surround features, respectively, and (b) their rotations [19].

representing the user actions to the system. For fast online analysis of the camera input we process the location and extent of the face, the locations of the eyes, the content of facial expressions, and the distribution of motion activity. Face detection was discussed in the previous section, the computation of the rest of the features is discussed next.

1) Face analysis: As discussed previously, face analysis starts with face detection. The presence of a face in the field of view of the camera is the main cue we use to arouse the system from its sleep mode. Future work can extend this easily by incorporating sound, such that a loud noise, or the utterance of a particular word can be used as triggers for activating the system.

The detection of eye locations and facial expression analysis both depend the detected face area. For the eye center localization, we used a technique based on isophote curvature, proposed by Valenti and Gevers [20]. The proposed method makes use of isophote properties to gain invariance to linear lighting changes (contrast and brightness) and rotational invariance. For every pixel, the center of the osculating circle of the isophote is computed from smoothed derivatives of the image brightness, so that each pixel can provide a vote for its own center. The eye center is surrounded by pixels whose curvature point in the eye-center direction, so it becomes very salient when these votes are pooled. The use of isophotes yields low computational cost (which allows for real-time processing) and robustness to rotation and linear illumination changes. Fig. 9 illustrates an example of the face and eye location on the feature analysis module.

The features extracted from the face allows for quantification of changes in different aspects. For instance the change in the scale of the facial area is indicative of movement towards the frame or away from it. The eye centers denote shifting foci of attention, although the system we employ does not have sufficient resolution to precisely determine the true focus of attention.

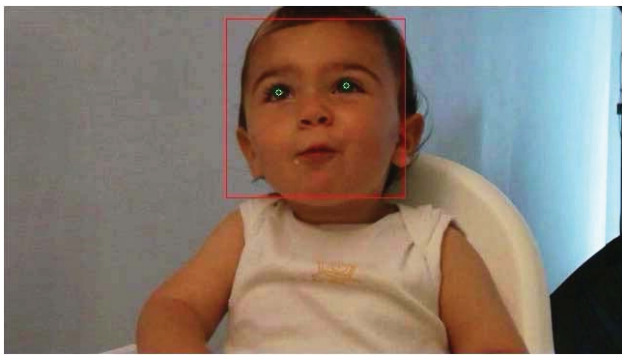
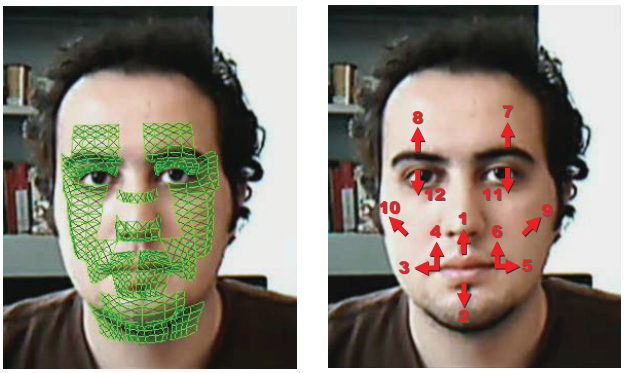


Fig. 9. An example of face and eye center localization.

For facial expression analysis we have used the system which is proposed in [6]. In this approach, the face is tracked by a piecewise Bézier volume deformation (PBVD) tracker, based on the



(a) (b)

Fig. 10. (a) The Bézier volume model. (b) The motion units.

system developed by Tao and Huang [21]. A three dimensional facial wireframe model is used for tracking. The generic face model consists of 16 surface patches, and it is warped to fit the estimated facial feature points, which are simply estimated by their expected locations with respect to the detected face region boundary. These expected locations are learned on a separate training set of faces.

The surface patches are embedded in Bézier volumes to generate a smooth and continuous model. A Bézier curve for $n+1$ control points can be written as:

$$x(u) = \sum_{i=0}^{n} b_i B_i^n(u),$$
$$x(u) = \sum_{i=0}^{n} b_i \binom{n}{i} u^i (1-u)^{n-i}, \tag{2}$$

where the control points b_i and $u \in [0,1]$ control model shape according to Bernstein polynomials, denoted with $B_i^n(u)$. The Bézier volume is an extension of the Bézier curve, and the displacement of the mesh nodes can be computed as $V = BD$, where B is again the mapping in terms of Bernstein polynomials, and D is a matrix whose columns are the control point displacement vectors of the Bézier volume.

After initialization of the facial model, head motion and facial surface deformations can be tracked. 2D image motions are estimated using template matching between frames at different resolutions. Previous frames are also used for better tracking. Estimated image motions are modelled as projections of true 3D motions. Therefore, 3D motion can be estimated using the 2D motions of several points on the mesh.

Expression classification is performed on a set of motion units, which indicate the movement of several mesh nodes on the Bézier volume with respect to the initial, neutral/frontal frame. 12 different motion units are defined as shown in Fig. 10. Unlike Ekman's Action Units [22], motion units represent not only the activation of a facial region, but also the direction and intensity of the motion. A naïve Bayes classifier is used to compute the posterior probabilities of seven basic expression categories (neutral, happiness, sadness, anger, fear, disgust, surprise).

2) Motion Energy and Activity Levels: The motion energy in a particular frame is computed by means of the optical flow. For its computation we use the technique proposed by Lucas and Kanade [17] for registration of images. This method assumes that the flow is essentially constant in a local neighbourhood of pixels under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood under a least squares criterion. By combining information from several nearby pixels, the Lucas-Kanade

method can often resolve the inherent ambiguity of the optical flow equation. It is also less sensitive to image noise compared to point-wise methods. In our particular case, we have used a pyramidal implementation of the Lucas-Kanade algorithm, developed by Jean-Yves Bouguet [23]. Fig. 11 shows a graphical example of the optical flow algorithm output for a particular frame.

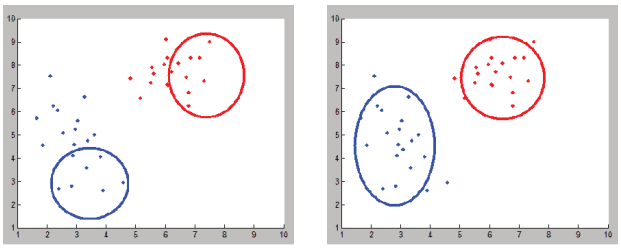
Fig. 11. Example of the optical flow vectors obtained in a frame using the pyramidal implementation of the Lucas and Kanade algorithm [23]. Optical flow vectors are represented as red arrows in the picture.

C. Learning and Adaptation

There are several ways to define interaction between a computer and its human user. The dominant paradigm is to specify the response of the computer precisely, given a certain input from the user. In the interactive photo frame, the manifestation of this paradigm is a static design of the system response logic, and a pre-specified input dictionary. There are however two immediate problems here. Our affect-sensing technology is not robust enough to assign crisp categories to different actions of different users. In other words, if the system is not trained for a specific person, there is a possibility that only a few input words will be activated during the lifetime of the system, and other response possibilities are left unexplored. The second problem is that the response dictionary of the system is not static, and grows each time a new video is added to the system.

The solution to both problems is to model the operation of the system as a dialogue, and let a consistent semiotics emerge through the interaction [24]. In this approach, the initial response of the system is random, or relates weakly to the actions of the user. However, during interaction, action-response pairs are stored. The system then periodically updates its response function by analysing the existing action-response pairs. This serves a two-fold purpose. 1) The response of the system becomes consistent over a period of usage, in that the user becomes able to trigger a certain response by a certain action, and these triggering actions are suitably idiosyncratic. 2) The system, by giving glyph-based feedback to the user, induces certain actions, yet if the user is not able to produce the expected valence, the learning process will shift the required activity to an appropriate level suitable for the user's activity range. In other words, the user and the system simultaneously adapt to each other, and for each user, the final response pattern of the system will be different.

Let F^t denote the feature responses collected during a session of interaction with a user. At a specific moment T of the session, if there are k active segments, and one additional segment that the user seeks to activate at the moment of analysis, there will be $k+1$ feature distributions, represented as $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, with $i = 1 \ldots k+1$. Here, each segment is activated by a feature response that is close to its distribution, as measured by the Mahalanobis distance between $\boldsymbol{\mu}_i$ and F^t.



(a) (b)

Fig. 12. The user responses (each point is one frame) projected to two dimensions. The response thresholds of the system are shown as ellipses for two segments (red and blue in the coloured version), (a) before adaptation (b) after adaptation.

We can take into account the idiosyncratic variations that are conditioned to users by letting the system adapt its response to the user. The terms that determine the system response are F^t, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. Since F^t is computed from the camera input recording user's behavior, the adaptation of the system is not concerned with it, but rather involves changing $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. The idea is to update these variables for an improved modeling of user behavior. Fig. 12 illustrates this idea on a toy example.

The procedure we use for improving the adaptation of the system is simple. At periodical intervals, the parameters of the system are updated as follows:

$$h_i(F^t) = \frac{p(F^t|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{j=1}^{k+1} p(F^t|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \quad (3)$$

$$\boldsymbol{\mu}_i' = \alpha\boldsymbol{\mu}_i + (1-\alpha)\frac{\sum_{t=1}^{T} h_i(F^t)F^t}{\sum_{t=1}^{T} h_i(F^t)}. \quad (4)$$

$$\boldsymbol{\Sigma}_i' = \alpha\boldsymbol{\Sigma}_i + (1-\alpha)\frac{\sum_{t=1}^{T} h_i(F^t)(F^t-\boldsymbol{\mu}_i)(F^t-\boldsymbol{\mu}_i)^T}{\sum_{t=1}^{T} h_i(F^t)}. \quad (5)$$

Here, $h_i(F^t)$ denotes the normalized membership probabilities of a particular set of features F^t for behaviour segment i, $p(F^t|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is computed from the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, and α is a control parameter. Small values of α will result in small adjustments in the systems behaviour, making it more responsive to the type of activities displayed by the user, as opposed to activities expected by the system. Large values of α may cause inconsistent behaviour in the system, and abrupt changes in response.

IV. System Assessment

We have constructed a working prototype of the system that has basic functionality. We summarize the achievements and assessment in this section.

A. Offline Segmentation

The offline segmentation module is completely implemented. To gain insight into its operation, we have manually segmented a number of video sequences. The system segmentation is then contrasted with manual segmentation, provided by five different persons for each video sequence. During manual segmentation, segments were also assigned labels. We have not constrained these labels in any way; the only constraint was conciseness. The freely available ANVIL multimedia annotation tool[4] was used.

Fig.13 shows a video sequence being processed in the ANVIL tool. Five different segmentations are displayed as rows at the bottom of

[4]http://www.anvil-software.de/

Fig. 13. The manual segmentation of videos and the corresponding automatically determined segmentation.

the video image. The temporal dimension is represented in each row in a left-to-right fashion. Labeled segments are represented as boxes, with the custom label written inside. The smoothed optical flow graph that is appended to the figure (aligned in the temporal axis) is not part of the annotation tool. It displays the result of automatic offline segmentation (as vertical bars) and the optical flow illustrates the 'reasoning' of the system in choosing these segments. The bars are elongated to intersect all five manual segmentations, so as to allow visual comparison. As it is evident from the figure, the most important segment boundaries (as evidenced by consensus among the taggers) is found by the automatic algorithm.

B. Real-time Feature Analysis

The real-time feature analysis module has been partly implemented. As we have discussed, some external software modules were employed to make the system work. The processing is no streamlined, and subsequently the computation burden of real-time feature computation is high. This is a common problem we have noted in similar systems. The SEMAINE API [25], which is developed for building emotion-oriented systems, and which provides a rich set of tools for this purpose, was considered for usage in an early stage of development. Our initial experiments have shown that enabling the facial feature analysis module in this system required a lot of computational resources. The information provided by the API in this modality is quite detailed, which led us to pursue a computationally cheaper system that would nonetheless be useful in guiding the interaction. The full assessment of this module is closely tied to usability studies with real subjects, which was not performed during the Workshop.

C. Real-time Facial Expression Analysis

We have assessed the accuracy of the eMotion software on the Cohn-Kanade AU-Coded Facial Expression Database [12]. In this database, there are approximately 500 image sequences from 100 subjects. These short videos each start with a neutral and frontal face display, and with little overall movement of the face display an emotional expression. Cohn-Kanade dataset has single action unit displays, action unit combinations, as well as six universal expressions, all annotated by experts. Without any manual facial landmark correction, the eMotion software provides 70.68 per cent average classification accuracy for six emotional expressions on this database. We have used 249 of the emotional expression sequences (46 joy, 49 surprise, 33 anger, 37 disgust, 41 fear, 43 sadness sequences) with three-fold cross validation to obtain the accuracy. Warping the generic face model of the eMotion software into a more accurate face representation anchored by seven manually annotated facial feature points (outer eye corners, inner eye corners, nose tip, and mouth corners) by a Thin-Plate Spline algorithm [26] has increased the average classification accuracy to 80.72 per cent. Fig. 14 shows the classification accuracy of the eMotion software for different emotional expressions, with and without manual landmark correction.

V. Conclusions and Future Work

We have developed a working prototype for an affect-responsive photo frame application. Our report sketches the main parts of the application, focusing on only visual features. The voice and speech modalities can be added to the system following the same principles, at the cost of higher computational complexity. We have completed the offline segmentation, feature analysis and the interface modules. The adaptation and dual-frame modules were not implemented during the Workshop. The dual-frame mode of the system is particulary interesting, as it solves the content acquisition and maintenance

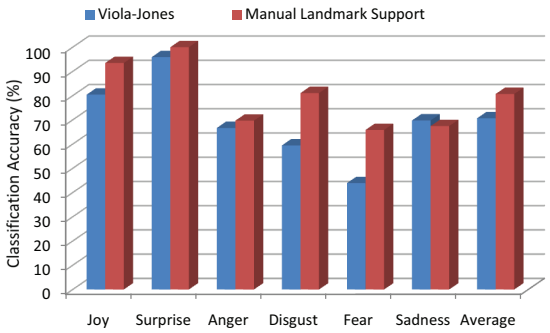


Fig. 14. Classification accuracies of eMotion software for different emotional expressions with and without manual landmark correction.

problems. This is the most important aspect that separates this work from similar digital constructions in the literature. We do not assume carefully recorded and annotated response patterns, but process the input and the output of the system automatically.

Our preliminary experiments have shown us that the proposed system is interesting and engaging. We have not conducted formal usability studies, but earlier prototypes were inspected and practical aspects of design were discussed. A thorough user assessment requires usability studies on a reasonable set of subjects, which can then reveal limitations of the system in longer term usage. It is conceivable that our automatic content management results in less meaningful segments than a hand-crafted set of responses. It remains to be seen whether the constant novelty created by the dual usage of the system is sufficient to offset this handicap, or even to turn it into an advantage.

REFERENCES

[1] M. Schröder, E. Bevacqua, F. Eyben, H. Gunes, D. Heylen, M. ter Maat, S. Pammi, M. Pantic, C. Pelachaud, B. Schuller, et al., "A Demonstration of Audiovisual Sensitive Artificial Listeners", in *Proc. Int. Conf. on Affective Computing & Intelligent Interaction, Amsterdam, Netherlands, IEEE*, 2009.

[2] T.H. Bui, J. Zwiers, M. Poel, and A. Nijholt, "Toward affective dialogue modeling using partially observable Markov decision processes", in *Proc. Workshop Emotion and Computing, 29th Annual German Conf. on Artificial Intelligence*, 2006, pp. 47–50.

[3] S. Agamanolis, "Beyond Communication: Human Connectedness as a Research Agenda", *Networked Neighbourhoods*, pp. 307–344, 2006.

[4] M. Mancas, R. Chessini, S. Hidot, C. Machy, R. Ben Madhkour, and T. Ravet, "Morface: Face morphing", *Quarterly Progress Scientific Report of the Numediart Research Program*, vol. 2, no. 2, pp. 33–39, 2009.

[5] N. Sebe, M.S. Lew, Y. Sun, I. Cohen, T. Gevers, and T.S. Huang, "Authentic facial expression analysis", *Image and Vision Computing*, vol. 25, no. 12, pp. 1856–1863, 2007.

[6] R. Valenti, N. Sebe, and T. Gevers, "Facial expression recognition: A fully integrated approach", in *Proc. 14th Int. Conf. of Image Analysis and Processing-Workshops*. IEEE Computer Society, 2007, pp. 125–130.

[7] R. Kaliouby and P. Robinson, "Real-time inference of complex mental states from facial expressions and head gestures", *Real-time vision for human-computer interaction*, pp. 181–200, 2005.

[8] J.N. Bailenson, E.D. Pontikakis, I.B. Mauss, J.J. Gross, M.E. Jabon, C.A.C. Hutcherson, C. Nass, and O. John, "Real-time classification of evoked emotions using facial feature tracking and physiological responses", *International journal of human-computer studies*, vol. 66, no. 5, pp. 303–317, 2008.

[9] B. Fasel and J. Luettin, "Recognition of asymmetric facial action unit activities and intensities", in *Int. Conf. on Pattern Recognition*, 2000, vol. 15, pp. 1100–1103.

[10] M. Pantic and L.J.M. Rothkrantz, "Automatic analysis of facial expressions: The state of the art", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424–1445, 2000.

[11] Y.L. Tian, T. Kanade, and J.F. Cohn, "Facial expression analysis", *Handbook of face recognition*, pp. 247–275, 2005.

[12] T. Kanade, J.F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis", in *Proc. AFGR*, 2000.

[13] D. Okwechime, E.J. Ong, and R. Bowden, "Real-Time Motion Control Using Pose Space Probability Density Estimation", in *Proc. ICCV*, 2009.

[14] P. Markopoulos, B. Bongers, E. Alphen, J. Dekker, W. Dijk, S. Messemaker, J. Poppel, B. Vlist, D. Volman, and G. Wanrooij, "The PhotoMirror appliance: affective awareness in the hallway", *Personal and Ubiquitous Computing*, vol. 10, no. 2, pp. 128–135, 2006.

[15] M. Zuliani, C. Kenney, and B. S. Manjunath, "A mathematical comparison of point detectors", *Computer Vision and Pattern Recognition Workshop*, vol. 11, pp. 172, 2004.

[16] J. Shi and C. Tomasi, "Good features to track", in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[17] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision", in *IJCAI*, 1981, pp. 674–679.

[18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 511–518.

[19] R. Lienhart and J. Maydt, "An extended set of haarlike features for rapid object detection", in *IEEE International Conference on Image Processing*, 2002, vol. 1, pp. 900–903.

[20] Roberto Valenti and Theo Gevers, "Accurate eye center location and tracking using isophote curvature", in *CVPR*, 2008.

[21] H. Tao and TS Huang, "Connected vibrations: a modal analysis approach for non-rigid motion tracking", in *Proc. CVPR*, 1998, pp. 735–740.

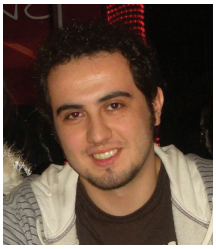
[22] P. Ekman, W.V. Friesen, and J.C. Hager, *Facial action coding system*, Consulting Psychologists Press Palo Alto, CA, 1978.

[23] Jean-Yves Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm", 2000.

[24] AA Salah and BAM Schouten, "Semiosis and the relevance of context for the ami environment", *Proc. European Conf. on Computing and Philosophy (ECAP)*, 2009.

[25] M. Schroeder, "The SEMAINE API: Towards a Standards-Based Framework for Building Emotion-Oriented Systems", *Advances in Human- Computer Interaction*, 2010.

[26] F.L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 6, pp. 567–585, 1989.



Hamdi Dibeklioğlu was born in Denizli, Turkey in 1983. He received his B.Sc. degree from Computer Engineering Department of Yeditepe University, in 2006, and his M.Sc. degree from Computer Engineering Department of Boğaziçi University, in 2008. He is currently a research&teaching assistant and a Ph.D. student at Intelligent Systems Lab Amsterdam, University of Amsterdam. His research interests include computer vision, biometrics, pattern recognition and intelligent human-computer interfaces. He works on Human Behavior Analysis under supervision of Professor Theo Gevers.
E-mail: h.dibeklioglu@uva.nl

Ilkka Kosunen is studying computer science at the University of Helsinki and also working as a research assistant at Helsinki Institute for Information technology, where he is developing various biosignal adaptive applications. His research interests include machine learning and biofeedback.
E-mail: ilkka.kosunen@hiit.fi

Marcos Ortega Hortas received his MSc degree in Computer Science from University of A Coruña, Spain, in 2004 and his PhD degree in 2009 from the Department of Computer Science of the same University, with a work focused on the use of retinal vessel tree as a biometric pattern for authentication purposes. He also worked on face biometrics studying the face evolution due to ageing effects as a visiting researcher on the University of Sassari in the Computer Vision Laboratory. He currently serves as a postdoctoral fellow on the University of A Coruña. His research areas of interest are medical image analysis, computer vision, biometrics and human behaviour analysis.
E-mail: mortega@udc.es

Albert Ali Salah received his PhD in 2007 from the Dept. of Computer Engineering of Boğaziçi University, with a dissertation on biologically inspired 3D face recognition. This work was supported by two FP6 networks of excellence: BIOSECURE on multimodal biometrics, and SIMILAR on human-computer interaction, which gave rise to the eNTERFACE Workshops. His research areas are human behaviour analysis, pattern recognition, biometrics, and multimodal information processing. He received the inaugural EBF Biometrics Research Award in 2006, and joined with the Signals and Images group at CWI, Amsterdam as a BRICKS scholar. He is presently a researcher at the Informatics Institute of the University of Amsterdam. He is the co-chair of the eNTERFACE'10 Workshop.
E-mail: a.a.salah@uva.nl

Petr Zuzánek was born in Trutnov, Czech Republic in 1988. He received his B.Sc. degree at Czech Technical University in Prague departure of Cybernetics in June 2010. He is M.Sc. student at Czech Technical University in Prague. He is currently working on the real-time tracking of nearly linear objects from video sequences captured by flying observer at Center for Machine Perception (http://cmp.felk.cvut.cz). This work is a continuation of his bachelor project. His supervisor is Dr. Karel Zimmermann.
E-mail: zuzanpet@fel.cvut.cz

Automatic Fingersign to Speech Translator

Pavel Campr[1], Erinç Dikici[2], Marek Hruz[1], Alp Kindiroglu[2], Zdenek Krnoul[1], Alexander Ronzhin[3], Hasim Sak[2], Daniel Schorno[4], Lale Akarun[2], Oya Aran[5], Alexey Karpov[3], Murat Saraclar[2], Milos Zelezny[1]

[1]University of West Bohemia, Czech Republic, [2]Bogaziçi University, Turkey, [3]SPIIRAS Institute, Russia, [4]STEIM, Netherlands, [5]Idiap Research Institute, Switzerland

***Abstract*— The aim of this project is to help the communication of two people, one hearing impaired and one visually impaired by converting speech to fingerspelling and fingerspelling to speech. Fingerspelling is a subset of sign language, and uses finger signs to spell letters of the spoken or written language. We aim to convert finger spelled words to speech and vice versa. Different spoken languages and sign languages such as English, Russian, Turkish and Czech are considered.**

***Index Terms*—fingerspelling recognition, speech recognition, fingerspelling synthesis, speech synthesis**

I. Introduction

The main objective of this project is to design and implement a system that can translate fingerspelling to speech and vice versa, by using recognition and synthesis techniques for each modality. Such a system enables communication with the hearing impaired when no other modality is available.

Although sign language is the main communication medium of the hearing impaired, in terms of automatic recognition, fingerspelling has the advantage of using limited number of finger signs, corresponding to the letters/sounds in the alphabet. Although the ultimate aim should be to have a system that translates the sign language to speech and vice versa, considering the current state of the art and the project duration, focusing on fingerspelling is a reasonable choice and provides insight to next coming projects to develop advanced systems. Moreover as fingerspelling is used in sign language to sign out-of-vocabulary words, the outcome of this project provides modules that can be reused in a sign language to speech translator.

The objectives of the project are the following:

- Designing a close to real time system that performs fingerspelling to speech (F2S) and speech to fingerspelling (S2F) translation
- Designing various modules of the system that is required to complete the given task.
  - Fingerspelling recognition
  - Speech recognition
  - Fingerspelling synthesis
  - Speech synthesis

II. System Overview

The overall system is implemented in client-server architecture. The modules are the client applications and are communicating through the server. The system is operating in close to real time. It takes the fingerspelling input from the camera, or the speech input from the microphone and converts it to synthesized speech or fingerspelling. The input and output can be selected among the supported languages for each module. The translation between different languages is handled via Google translate APIs. The system flowchart can be seen in Fig.1.

A simple game scenario is defined as follows, for demonstration purposes:

SP- Hi, I am Alexander, from Russia
FS- Hi, I am Alp, from Turkey
SP- Do you want to play city names game?
FS- Yes
SP- Ok, I start. London
FS- Naples
SP- St. Petersburg
FS- Grenoble
SP- ...

III. Literature Survey

A. Fingerspelling recognition

The fingerspelling recognition task involves the segmentation of fingerspelling hand gestures from image sequences. Through the classification of features extracted from these images, sign gesture recognition can be achieved. Since a perfect method of segmenting skin color objects from images with complex backgrounds has not yet been proposed, recent studies on fingerspelling recognition make use of different methodologies. Liwicki and Everingham [1] focuses on the segmentation of hands by

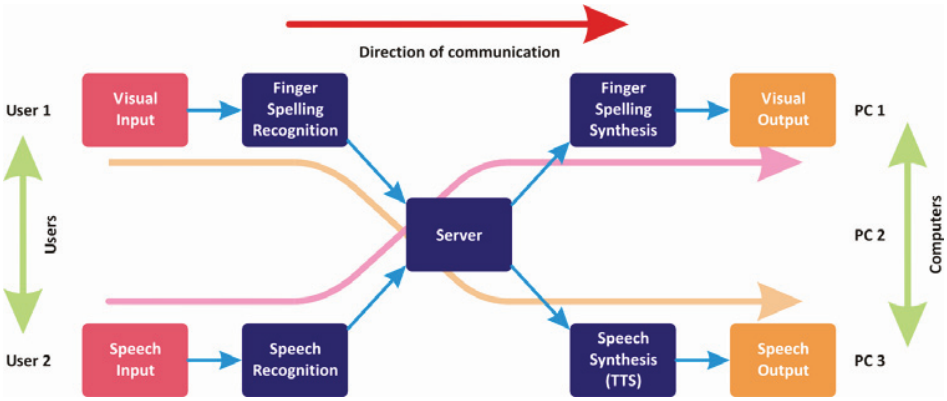


Fig. 1. System flowchart

skin color detection methods and background modeling. Then, Histogram of Oriented Gradient descriptors are used to classify hand features with Hidden Markov Models. Goh and Holden [2] incorporate motion descriptors into skin color based segmentation to improve the accuracy of hand segmentation. Gui et al. [3] makes use of human past behavioral patterns in parallel with skin color segmentation to achieve better hand segmentation.

B. Fingerspelling synthesis

The goal of an automatic sign language synthesizer is the reproduction of human behavior during the signing. The sign language synthesizer should express manual components (position and shape of hands) as well as non-manual components (face expression, lip articulation etc.) of the performed signs. In the general task, sign language synthesis is implemented in several steps. Firstly the source utterance has to be translated into the corresponding sequence of signs since the sign language has different grammar than the spoken one. Then the relevant signs have to be concatenated to continuous utterance. The non-manual components should be partially supplemented by a talking head system that is able to articulate the utterance or express the face gestures.

The straightforward solution of sign language synthesis is based on video records of signing human. A concatenation of these records has very good quality and realism. On the other hand, we can find an avatar animation allowing low-bandwidth communication, arbitrary 3D position and lighting, and to change an appearance of the animation model.

There are two main solutions how to solve the task of sign language synthesis via avatar animation. The first one is based on the recording and reproduction of signing speaker in the 3D space using motion capture [4]. The second one is more artificial and is based on a symbolic notation of signs [5][6].

The first solution is to use the recorded data to control the animation of the avatar directly. The advantages are 3D trajectories of whole body and very realistic motions of the animation model. Low accuracy and extensibility of recorded signs are considered as disadvantages as well as the need of special and expensive equipment. Advantages of synthesis from the symbolic notation are high accuracy of the generated trajectories, relatively easy editing of symbols, and a possibility to add new features. A lexicon for the synthesis can be collected from different sources and created at different times. The disadvantages are a complicated conversion of symbols to animation trajectories and authenticity of final animation.

C. Speech recognition

Human speech refers to the processes associated with the production and perception of sounds used in spoken language, and automatic speech recognition (ASR) is a process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a software or hardware module. Several kinds of speech are identified: spelled speech (with pauses between letters or phonemes), isolated speech (with pauses between words), continuous speech (when a speaker does not make any pauses between words) and spontaneous natural speech in an inter-human dialogue. The most common classification of ASR by recognition vocabulary is the following [7]:

- small vocabulary (10-1000 words);
- medium vocabulary (up to 10 000 words);
- large vocabulary (up to 100 000 words);
- extra large vocabulary (up to and above million of words that is adequate for inflective or agglutinative languages)

Recent automatic speech recognizers exploit mathematical techniques such as Hidden Markov Models (HMM), Artificial Neural Networks (ANN), Bayesian Networks or Dynamic Time Warping (dynamic programming) methods. The most popular ASR models apply speaker-independent speech recognition, though in some cases (for instance, personalized systems that have to recognize owner only) speaker-dependant systems are more adequate.

In framework of the given project a multilingual ASR system is constructed applying the Hidden Markov Model Toolkit (HTK version 3.4) [8]. Language models based on statistical text analysis and finite-state grammars are

implemented for ASR of continuous phrases or messages [7].

D. Speech synthesis

Speech synthesis is the artificial production of human speech. Speech synthesis (also called text-to-speech (TTS) system converts normal orthographic text into speech translating symbolic linguistic representations like phonetic transcriptions into speech. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database (compilative, HMM-based or unit selection speech synthesis methods) [9]. Systems differ in the size of the stored speech units; a system that stores allophones or diphones provides acceptable speech quality but the systems that are based on unit selection methods provide a higher level of speech intelligibility. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create voice output. The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood (intelligibility).

E. Properties of the considered languages

The Czech, English, Russian and Turkish languages are included in the speech scope of the system and the Czech, Turkish and Russian fingerspelling alphabets are included in the visual scope.

Turkish is an agglutinative language with relatively free word order. Due to their rich morphology Czech, Russian and Turkish are challenging languages for ASR. Recently, large vocabulary continuous speech recognition (LVCSR) systems have become available for Turkish broadcast news transcription [10]. An HTK based version of this system is also available. LVCSR systems for agglutinative languages typically use sub-word units for language modeling.

The Russian language belongs to the Slavonic branch of the Indo-European group of languages, which are characterized by a tendency to combination (synthesizing) of a lexical morpheme (or several lexical morphemes) and one or several grammatical morphemes in one word-form. So, Russian is a synthetic inflective language with a complex mechanism of word-formation. For large vocabulary Russian ASR, it is required to apply a recognition vocabulary in several orders larger than for English or French ASR because of existence of prefixes, suffixes and endings that essentially decreases both accuracy and speed of recognition. Zaliznjak's grammatical dictionary of Russian contains above 150 thousand words and due to word-formation rules it allows to extract all the dictionary entries and to obtain over two million various word-forms. For instance, verbs can generate up to two hundred word-forms, which have to be taken into account at speech recognition. Besides, most word-forms of the same word differ in endings only, which are pronounced in continuous speech not as clearly as beginning parts of words. Misrecognition in endings results in misrecognition of word and whole sentence because of word discordance.

Moreover, word order in Russian sentences is not restricted by hard grammatical constructions, like in English or German that complicates creation of statistical language models or essentially decreases their effectiveness. N-gram language models for Russian are larger in orders in contrast to English and have perplexity and entropy estimations in three-four times higher.

SAMPA phonetic alphabet for the Russian language includes 42 phonemes: 36 units for consonants and six for vowels, so consonants ambiguity is rather high in Russian. An automatic phonetic transcriber is required for creation of the recognition vocabulary for Russian ASR. Rules for transformation from orthographic text to phonemic representation are not very complicated for Russian; however, the main problem is to find position of stress (accent) in a word-form. There exist no common rules to determine stress positions; moreover, compound words may have several accents at once. Only knowledge-based approaches can solve this challenge.

The three different fingerspelling alphabets included in the system contain varying characteristics that make their combined recognition a challenging problem. The Turkish Fingerspelling Alphabet (TFA), seen in Fig.2, contains seven gestures performed by one hand and twenty two gestures performed by two hands.

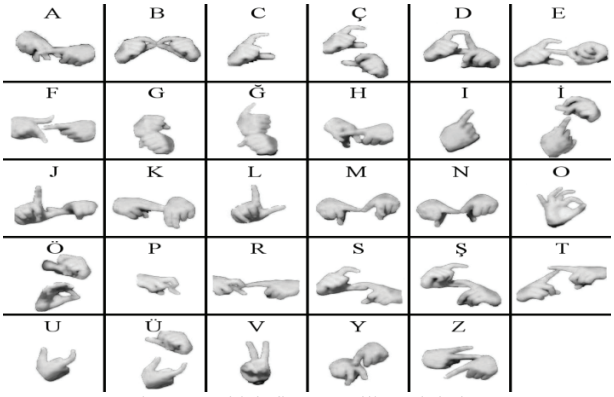


Fig. 2. Turkish fingerspelling alphabet

Fig. 3. Russian fingerspelling alphabet

In contrast to the Turkish alphabet, all signs of the Russian fingerspelling alphabet (Fig.3) are performed by one (right) hand. The Czech sign language contains a one

handed and a two handed sign for each letter. Therefore, this combination of gestures creates a need to handle the processing of different kinds of letters separately.

IV. Client-server architecture

Since the aim of this project is to help the communication of two people, the use of a computer network is essential to allow remote communication.

As seen in Fig.1, the whole communication system has two input parts (one for each user), the central server and two output parts (again, one for each user).

The central server, located outside of the user computers, runs a standalone application, which communicates with the applications located in the users' computers (one input and one output application for every user). As these applications connect to the server we can call them *clients*.

The server has these features:

- can handle multiple discussions ("sessions"), i.e. multiple user pairs can discuss separately on the same server
- receives text messages from input clients
- stores all messages
- translates messages to another languages using Google Translate API
- sends messages to output clients

The "message" can be a single letter (e.g. received from a finger spelling recognition client), a single word or a sentence (e.g. from a speech recognition client). The server automatically concatenates single letters into words and single words into sentences.

The server is implemented as a web server that receives and delivers content using the HTTP (Hypertext Transfer Protocol) over the Internet. The server receives requests from the clients and sends a response back.

Example to retrieve messages by an output client:

```
http://[server url]/dialogue?list=all&session=tom_and_bob&format=json&translate_lang=cs_CZ
```

This lists all messages from "tom_and_bob" session, translates all messages into "cs_CZ" language and sends the response in JSON format.

Example to send a new message by an input client:

```
http://[server url]/dialogue?language=en_GB&user_id=Tom&sentence=Hello+world&session=tom_and_bob
```

This adds the new sentence "Hello world" in "en_GB" language by user "Tom" into "tom_and_bob" session.

The advantage of this client-server architecture is the possibility to have multiplatform client applications created in any programming language which supports HTTP communication.

V. Fingerspelling recognition

The recognition system is based on video input without any markers on the hands. The supported languages are Turkish fingerspelling and Czech fingerspelling. Some stop signs are defined as well for end of word and end of sentence.

A. Dataset Collection

For the purposes of this project, we have collected a fingerspelling database from 11 different users. Five of these users perform Turkish fingerspelling while four perform Czech and two perform Russian. Of the 11 users, one Turkish and three Czech users are native signers who are hearing impaired.

Fig. 4. Turkish fingerspelling database

The videos are recorded by a mini-dv camera at 25fps at a resolution of 640x480. The signs are performed in front of a black background using a constant camera distance and angle. The signers wear dark colored clothes with long sleeves. As each letter is repeated five to eight times, the total length of the database for each signer changes between 30 and 45 minutes. Due to different recording environments lighting conditions and camera calibration settings slightly vary from subject to subject. Some samples from Turkish fingerspelling database is shown in Fig.4.

B. Hand Tracking and Segmentation

The highly mobile and self occluding nature of hands makes tracking hand gestures a challenging task. While signing in a natural manner, hands often tend to interact with each other, cross over the face and make movements that are sudden and rapid. In order to handle as many of the exceptional cases without using too much computational time, we used a tracking algorithm based on the Continuously Adaptive Meanshift (Camshift) method. The Camshift algorithm is a semi automatic tracking algorithm that accomplishes tracking by using the color properties of tracked objects. It is a robust non-parametric tracking algorithm that converges on the peaks of a probability distribution image [11].

As the original Camshift algorithm is designed for face tracking, it shows a few inadequacies in continuous fingerspelling recognition tasks. We replace the manual initiation by implementing a motion based hand detection module. The module requires the user to wave his hands for a few seconds before commencing signing to generate a person specific color histogram.

Following this method, histogram generation is performed. To negate the effects of background pixels

present in the histogram, the Weighted Histogram and Ratio Histogram techniques suggested in [11] are performed.

During tracking, we handle issues caused by simultaneous two hand tracking and tracking failures with a hierarchical hand redetection module [12]. With this module, by marking the combination and separation of tracking boxes, we keep track of the number of hands in the search boxes.

The tracking is performed on reduced resolution video for faster performance. Following tracking, masks of obtained hands are segmented from original sized images using color and motion cues.

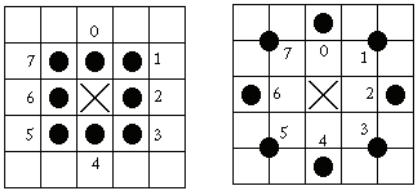
C. Feature Extraction

On the segmented hand images, we used the following shape descriptors as features to mathematically represent our hand images.

1) Local Binary Patterns

Local Binary Patterns (LBP) were introduced by Ojala [13] for texture representation. The LBP is used across various computer vision fields (e.g. image synthesis, light normalization, face detection, face/expression recognition). It has been successfully used for hand detection in cluttered images [14]. We use LBP for hand shape description.

First a LBP image is computed. The algorithm moves a defined patch along all the pixels in an image. The evaluated pixel is in the center of the patch. Depending on the size and shape of the patch, the resulting LBP image changes. We use a circular 8-neighbourhood patch with radius one and two pixels. The patches can be seen in Fig.5. If the brightness of a pixel in the patch is greater or equal to the evaluated pixel's brightness we assign a binary label 1 to the proper location in the patch. If the patch brightness is lower than the evaluated brightness we assign label 0 to it. In each patch we evaluate eight locations (or combinations of locations) which yields an 8-bit number. This number is assigned to the location of the evaluated pixel and the patch moves to the next pixel. Next, we compute a histogram of the LBP image, which we use as a feature vector. For normal LBPs there are 256 histogram bins, each bin for one pattern. In practice it has been shown that all the patterns are not important for recognition. Most of the information is in the patterns that have two or less changes between 0 and 1 in its binary representation. Such LBPs are called uniform [15].



(a) (b)

Fig. 5. Examples of LBPs. We use LBP with radius one (a) and two (b). Numerical values represent the position of the patch in the binary representation of the pattern.

There exist 58 such patterns and all the other patterns are moved to the 59th bin of the histogram. This means that for uniform LBP the feature vector is of size 59.

We implemented both uniform and non-uniform LBP both with the patch radius one and two.

2) Hu Moments

Used in numerous computer vision applications as shape descriptors, the invariant moments of Hu are calculated from central image moments using the formulas below [16]:

$I_1 = \eta_{20} + \eta_{02}$

$I_2 = (\eta_{20} - \eta_{02})^2 + (2\,\eta_{11})^2$

$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$

$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$

$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$

$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

Calculated from binary shape masks, we use the seven Hu moments as rotation, scale and translation invariant feature vectors.

3) Elliptic Fourier Descriptors

Elliptic Fourier descriptors on shape signatures are widely used for shape analysis and recognition [17][18]. These descriptors represent the shape of the object in the frequency domain. The lower frequency descriptors contain information about the general features of the shape, and the higher frequency descriptors contain information about finer details of the shape. Although the number of coefficients generated from the transform is usually large, a subset of the coefficients is enough to capture the overall features of the shape.

Consider an *n*-harmonic elliptic Fourier descriptor representation of any 2-D curve, namely:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{N} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos(kt) \\ \sin(kt) \end{bmatrix}$$

where (a_0, c_0) the center of the curve and (a_k, b_k, c_k, d_k), $k = 1,2,\ldots n$ are elliptic Fourier coefficients of the curve up to *n* Fourier harmonics.

The Euclidean invariants can be defined as follows:

$$A_k^2 = \frac{I_k + \sqrt{I_k^2 - 4J_k^2}}{2}$$

$$B_k^2 = \frac{J_k^2}{A_k^2}$$

where $I_k = a_k^2 + b_k^2 + c_k^2 + d_k^2$ and $J_k = |a_k d_k - b_k c_k|$. Here A_k and B_k are the major and minor axis lengths of the k^{th} ellipse fit to the shape.

4) Radial Distance Function

The radial distance function method, presented in [19] is a contour based method. Using the distance of a seed-point (possibly the center of mass) in all directions to the closest background pixel (Fig.6), a description of the image is given as a feature vector. The calculated image descriptors are invariant to translation, size and rotation. Rotation invariance is achieved by choosing the angle with the smallest radial distance as the point for each seed point.

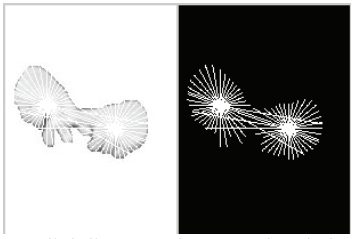


Fig. 6. Radial distances for a two-handed gesture

While describing an isolated hand, the radial distance function is an efficient measure as it is possible to represent finger locations and notable extensions of the hand. However, when describing blobs consisting of not completely overlapping, but touching hands, obtaining a point which has a straight line distance to both hands may not be possible. For this reason, we attempt to find the centers of gravity belonging to both hands. By using the image moments, we calculate the parameters of the smallest ellipse that covers both hands using the formulas below.

$$a = \frac{M_{20}}{M_{00}} - x^2 \qquad b = 2\left(\frac{M_{11}}{M_{00}} - x_c\, y_c\right)$$

$$c = \frac{M_{02}}{M_{00}} - y^2$$

$$l_1 = \frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}$$

$$l_2 = \frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}$$

$$\theta = \frac{1}{2} tan^{-1}\left(\frac{b}{a-c}\right)$$

In the calculations, l_1 and l_2 are the principal axes of the bounding ellipse centered on the centroid of the image and Θ is their rotation angle. By dividing the image using the minor axis l_2 as a separator, we effectively divide the blob into two approximately equal parts belonging to different hands. After calculating the centroid of each part using image moments, we obtain seed locations for two radial distance functions that are sufficient to describe a hand blob consisting of two hands, as seen in Fig.6.

D. Recognition and Results

For gesture recognition, we make use of a two layered classification method. First, we decide whether the given frame is a keyframe where a gesture is being performed or a transition frame where the hands are moving from one gesture to another. Then, only if the given frame is a key-frame we try to classify the hand gesture in the current keyframe into one of the previously given classes.

1) Keyframe Selection

In a hand gesture recognition setup, a hand gesture can either be represented with a single frame or through the usage of multiple frames in a sequence. Since we represented each hand gesture through a single static snapshot of itself, motion of hands, together with some image quality features such as motion blur provides satisfactory results in classifying a gesture as either a key or a transition frame. We make use of an unsupervised classification method for keyframe selection using three features namely the global motion of the hands, the amount of change in hand contours and the presence of motion blur.While signing consecutive gestures, the signer first moves her/his hands to a certain start position, moves her/his hands to perform the gesture and then either starts moving on to the next gesture or the start position. Therefore, while performing the gestures, she/he pauses for brief moments either while starting or completing the performed gesture and moving on to the next one. The use of motion and external hand contour change thus attempts to mark the moment where the motion of hands has slowed down in keyframe selection.

In addition, we also search for the presence of image blur around the external contours of the hand to eliminate any images where a deformation in the hand shape image may yield to a false recognition.

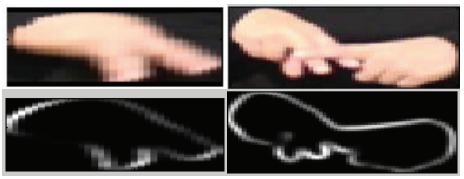


Fig. 7. Motion blur example

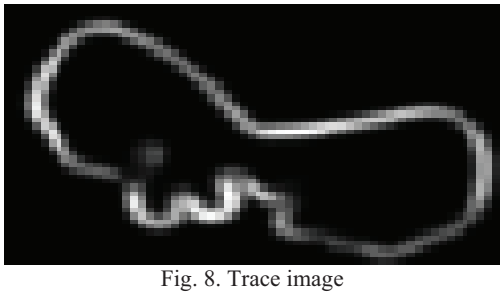


Fig. 8. Trace image

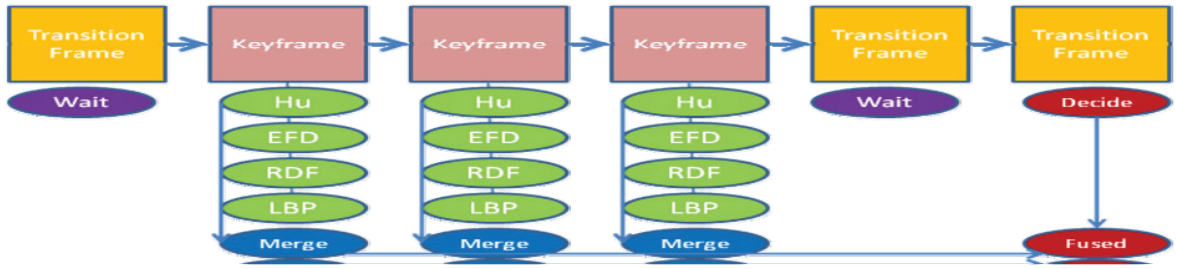


Fig. 9. Classification pipeline

Compared to unblurred images, a major characteristic of blurred images is that edges tend to be smoother and contain smaller gradient values (Fig.7). Therefore, focusing on the distribution of gradient values in a certain image patch can give us an idea about the presence of partial motion blur.

Using the image derivative masks,

$$Dx = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad Dy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

we convolve the images to obtain the derivative images Ix and Iy. From those images, using a Gaussian window, we obtain the smoothed squared image derivatives Ix^2 and Iy^2. Then we calculate the trace of the image in the same manner that is used to calculate the image trace for the Harris corner detector (Fig.8).

$$A = k * (Ix^2 2 + Iy^2)^2$$

As the trace image yields the most significant results in the edge areas that separate the hands from the background, we compute the trace image for small windows around the hand contour. We capture nxn sized small images around the hand contours that are at least $n^2/4$ pixels apart from each other (n=7). Then since we are looking for motion blur which is nonexistent in the direction of motion, the magnitude of the difference of the maximum and minimum gradient strength values of the same image can be used to hint an increase or decrease in the amount of motion blur [20].

2) Hand Gesture Recognition

In hand gesture classification hand gesture features of different types are handled individually and are classified using the K-nearest neighbor algorithm. For each recognized keyframe, a decision is obtained by fusing the classification results from Hu moments (Hu), Elliptic Fourier Descriptors (EFD), Radial Distance Functions (RDF) and Local Binary Patterns (LBP).

The results from the underlying features are then fused using weighted majority voting to obtain a keyframe based decision.

As gestures belonging to successive keyframes are classified, they are gathered until a certain number of successive transition frames appear. The gathered classification results of different keyframes are then fused together using majority voting (Fig.9).

For offline testing, we have used the fingerspelling videos belonging to the Turkish fingerspelling alphabet. Images belonging to each subject in the dataset have been divided into equal sized training and test sets (740 videos each). The recognition data for each subject's test videos are tested in two settings; one which includes the user's own training data and one that excludes it. The results of individual classification are shown in Table I and anonymous classification are presented in Table II.

TABLE I
TFL USER RECOGNITION ACCURACY

	Hu	EFD	RDF	LBP	Fused
Subject1	0.53	0.63	0.67	**0.85**	0.88
Subject2	0.56	0.70	0.78	**0.91**	0.93
Subject3	0.54	0.75	0.77	**0.87**	0.87
Subject4	0.48	0.61	0.61	**0.74**	0.78
Subject5	0.59	0.79	0.76	**0.94**	0.96
Average	0.54	0.70	0.72	**0.86**	0.88

TABLE II
TFL ANONYMOUS USER RECOGNITION ACCURACY

	Hu	EFD	RDF	LBP	Fused
Subject1	0.31	0.15	**0.39**	0.28	0.43
Subject2	0.23	0.06	**0.24**	0.21	0.28
Subject3	0.36	0.30	**0.48**	0.29	0.45
Subject4	0.07	0.28	**0.61**	0.48	0.54
Subject5	0.32	0.20	**0.45**	0.23	0.40
Average	0.26	0.20	**0.43**	0.30	0.42

As it can be inferred from the average results, having the user's own data in the recognition set doubles the accuracy of recognition from 42% to 88%. The main reasons of this difference can be inferred as the variances in illumination and the minor differences in the performances of the signers.

However, since the overall aim of the system is to provide an online word level recognition system, such accuracy rates on their own were deemed insufficient. For that reason we opted to implement a vocabulary list of 2000 words and match the signed letters to the closest word using Levenshtein distance [21]. We compared the sequence of recognized letters when the user returns to rest position (hands separated and lowered to opposing sides of the body), to each word using a normalized Levenshtein distance. As a result, the online system was able to send

word level messages to the server with a much greater accuracy.

VI. SPEECH RECOGNITION

We have implemented two different speech recognition modules. First is continuous speech recognition (implemented for Turkish) and the other is spelled speech recognition (implemented for Russian and English).

A. Continuous Speech Recognition

Automatic speech recognition (ASR) is needed in the first stage for the one-way communication from a speaking person to a hearing impaired person by converting spoken words to text that gets synthesized to sign language in later stages. We integrated a Weighted Finite-State Transducer (WFST) based large-vocabulary continuous speech recognition system developed at Boğaziçi University into this multimodal communication platform [22][10]. The integrated system is currently capable of recognizing just Turkish utterances since language and acoustic models were readily available only for Turkish.

The morphologically productive languages such as Turkish, Finnish, and Czech present some challenges in ASR systems. The out-of-vocabulary (OOV) rates for a fixed vocabulary size are significantly higher in these languages. The higher OOV rates lead to higher word error rates (WERs). Having a large number of words also contributes to high perplexity numbers for standard *n*-gram language models due to data sparseness. Turkish, being an agglutinative language with a highly productive inflectional and derivational morphology is especially prone to these problems.

The speech recognition problem is treated as a transduction from input speech signal to a word sequence in the WFST framework [23]. The WFSTs provide a unified framework for representing different knowledge sources in ASR systems. A typical set of knowledge sources consists of a transducer H modeling context-dependent phones as hidden Markov models (HMMs), a context-dependency network C transducing context-dependent phones to context-independent phones, a lexicon L mapping context-independent phone sequences to words, and an n-gram language model G assigning probabilities to word sequences. The composition of these models $H \circ C \circ L \circ G$ results in an all-in-one search network that directly maps HMM state sequences to weighted word sequences, where weights can be combinations of pronunciation and language model probabilities. The WFST also offers finite-state operations such as *composition*, *determinization* and *minimization* to combine all these knowledge sources into an optimized all-in-one search network.

The morphology as another knowledge source can be represented as a WFST and can be integrated into the WFST framework of an ASR system. The lexical transducer of the morphological parser maps the letter sequences to lexical morphemes annotated with morphological features [24]. The lexical transducer can be considered as a computational dynamic lexicon in ASR in contrast to a static lexicon. The computational lexicon has some advantages over a fixed-size word lexicon. It can generate many more words using a relatively smaller number of root words (55,278) in its lexicon. So it achieves lower OOV rates. In the WFST framework, the lexical transducer of the morphological parser can be considered as a computational lexicon M replacing the static lexicon L. Since M outputs lexical morphemes, the language model G should be estimated over these lexical units. Then with the morphology integrated, the search network can be built as $H \circ C \circ M \circ G_{morpheme}$. The decoding for the best path in the resulting network is a single-pass Viterbi search.

Note that the word-based models output word sequences such as "merhaba saat on üç haberleri ajanstan alıyorsunuz", while the morphology-integrated model outputs lexical morpheme sequences such as "merhaba[Noun] saat[Noun] on[Adj] üç [Adj] haber[Noun] +lAr[A3pl] +SH[P3sg] ajans[Noun] +DAn[Abl] al[Verb] +Hyor[Prog1] +sHnHz[A2pl]". Therefore, we use the morphological parser as a word generator to convert the recognition output to words.

We evaluated the performance of the speech recognition system on a Turkish broadcast news transcription task. The acoustic model uses hidden Markov models (HMMs) trained on 188 hours of broadcast news speech data [10]. In the acoustic model, there are 10843 triphone HMM states and 11 Gaussians per state with the exception of the 23 Gaussians for the silence HMM. The test set contains 3.1 hours of speech data that has been pre-segmented into short utterances (2,410 utterances and 23,038 words). We used the geometric duration modeling in the decoder.

The text corpora that we used for estimating the parameters of statistical language models are composed of 182.3 million-words BOUN NewsCor corpus collected from news portals in Turkish [24] and 1.3 million-words text corpus (BN Corpus) obtained from the transcriptions of the Turkish Broadcast News speech database [10].

As a baseline word language model, we built 200K vocabulary 3-gram language model. Our previous study showed that higher vocabulary sizes than 200K and higher n-gram orders did not improve the accuracy significantly [10]. The OOV rate for 200K word vocabulary is about 2% on the test set.

For the morphology-integrated model, the optimal n-gram order of the language model over lexical morphemes was chosen as four. The OOV rate of the morphological parser is 0.68% on the test set.

Fig.10 shows the word error rate versus run-time factor for 200K vocabulary word model Word-200K and the morphology-integrated model MP. The improvement in OOV rate for the morphology-integrated model translates to WER reductions.

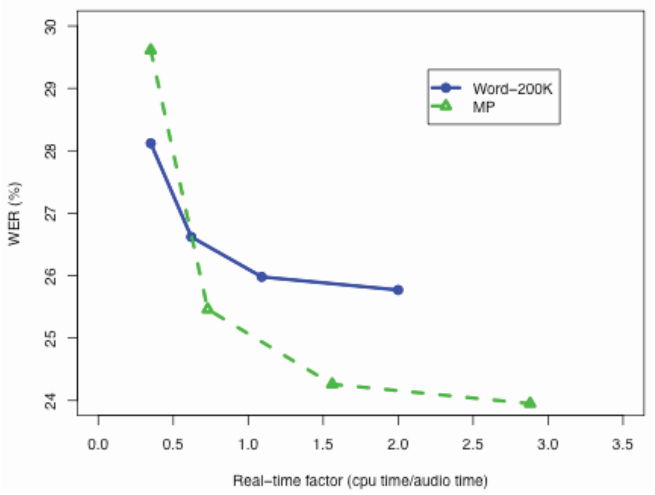


Fig. 10. Word error rate versus real-time factor obtained by changing the pruning beam width from nine to 12

For this communication platform, we implemented a voice activity detection (VAD) system to prevent false triggers and improve recognition accuracy. A binary supervised classification methodology has been adopted for this purpose. The "*speech*" class is trained with improvised talk and readings on a silent background and the "*nonspeech*" class contains silence, noise and some other noisy activities (cough, tapping on the microphone, mouse clicks, etc.). We use 13 dimensional MFCC vectors as features and GMMs with 16 components for training. Testing is done in an online fashion and the decision is given by the likelihood ratio test.

B. Spelled Speech Recognition

Spelled speech input (letter-by-letter input) is widely used by humans for rare and out-of-vocabulary words (for instance, personal names, city names, e-mail addresses, etc). A speaker-dependent automatic speech recognition (ASR) system was developed and embedded into the global system.

A single stationary microphone located 30-40 cm from the speaker's mouth is used for speech input. As acoustic features we used 13-dimensional Mel-Frequency Cepstral Coefficients (MFCC), including 0-th coefficient, with the first and second derivatives calculated from 26 channel filter bank analysis of 20 ms long frames with 10 ms overlap. Thus, the frequency of audio feature vectors is 100 Hz. Cepstral Mean Subtraction is applied to audio feature vectors.

Acoustic modeling and recognition of phonemes and letters of the recognition vocabulary are based on Hidden Markov Models (HMM). The acoustical models are realized as HMMs of the context-independent phones with mixture Gaussian probability density functions (GMMs). HMMs of phones have three meaningful states (and two additional states intended for concatenation of the phones in the letter models), see Fig.11.

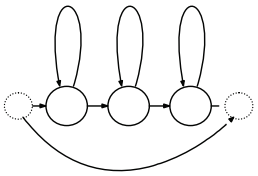


Fig. 11. Topology of HMM-based acoustical model for a phoneme

Fig.12 shows an example of complex HMM-based model for the isolated word "seven". One can see that there are five phones in this word and the second phone /e/ may sometimes disappear in pronunciations of some people.

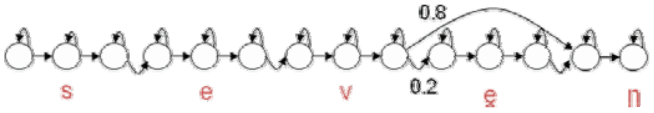


Fig. 12. Topology of HMM for the isolated word "seven"

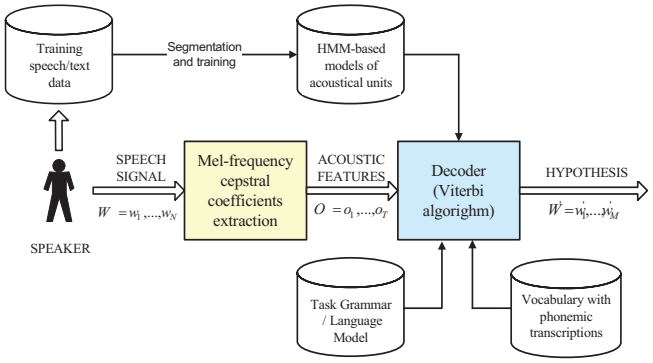


Fig 13. Architecture of spelled speech recognition system

Developed ASR system is multilingual and able to recognize letters spelled both in English and in Russian. The lexicon of ASR contains 26 English letters, plus 31 Russian letters (there exist 33 letters in the Russian language, but we employed 31 of them only, because two letters - the soft sign "Ь" and the hard sign "Ъ" have not their own phonetic representation, but an influence on previous letter's pronunciation in continuous speech only), plus digits for both languages looped in the null-gram model. Moreover, two system commands were additionally introduced in the system: "DEL" (backspace) – to delete the last said but misrecognized letter or word; "DOT" (point) – to indicate the end of the sentence input. English and Russian vocabularies use joint pool of trained phone models. A pause (silence) between two letters' input that lasts more than five seconds means the "SPACE" symbol. The general architecture of spelled speech recognition system is shown in Fig.13, one can notice that there are two main work modes: system training and speech decoding.

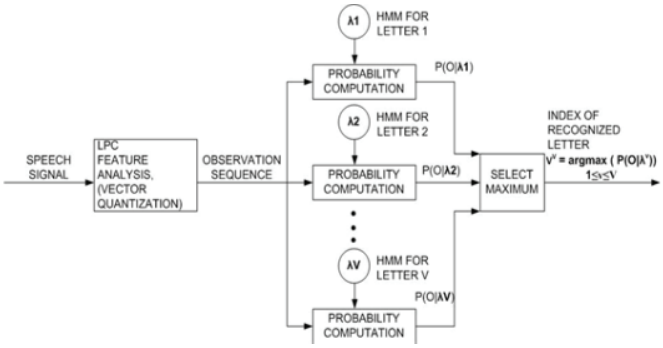


Fig. 14. A general algorithm of spelled speech decoding

The stage of system training includes the following steps:

- manual transcription of a lexicon of an applied domain;
- creation of a finite-state grammar of an applied domain;
- preparation of a training speech corpus;
- coding the speech data (feature extraction);
- definition of topology of HMMs (prototypes);
- creation of initial HMMs for phones list by the flat start;
- re-estimation of HMMs parameters of monophones using a labeled speech corpus and Baum-Welch algorithm;
- mixture splitting.

In order to train the speech recognizer a speech corpus was recorded in office conditions using the distant talking and directed microphone. Totally we have recorded about 20 minutes of speech data from one speaker, these data were labeled semi-automatically in the terms of phones.

The isolated speech decoder (see Fig.14) uses Viterbi-based token passing algorithm [25]. The input phrase syntax is described in a simple grammar loop that allows recognizing one vocabulary item in a hypothesis. The audio speech recognizer operates very fast (less than 0.1xRT) so the result of speech recognition is available almost immediately after detection of speech end by the activity detector.

The performance of ASR was evaluated by another speech data, collected in the same office conditions as the training part. Training and testing databases for the automatic speech recognition system was recorded in one session and each letter of both languages was said 20 times for the training purpose and 10 more times for the system evaluation and testing. Tables III and IV show the accuracy of speech recognition (in the form of confusion matrices) for English and Russian letters, correspondingly. In these confusion matrices, the sign "+" denotes a 100% recognition rate (10 instances of 10) and "-" means 0% (0 instances recognized of 10 pronounced). Most of the letters are recognized with 100 % rate; however, some letters (for example, English consonant letters B /b' i/ and D /d' i/ or vowel letters A /e i/ and I /a i/) are highly confused. The recognition rate (accuracy) for all the English letters was 93.1% on average, and 90% for the Russian letters. English spelled speech is recognized a bit better than the Russian one because of the smaller alphabet size.

TABLE III
CONFUSION MATRIX FOR ENGLISH LETTERS RECOGNITION

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D	-	5	-	3	1	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	9	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
I	2	-	-	-	-	-	-	-	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
K	1	-	-	-	-	-	-	-	-	-	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
L	-	-	-	-	-	-	-	-	-	-	-	7	3	-	-	-	-	-	-	-	-	-	-	-	-	-
M	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-
N	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-
O	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-
P	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	8	-	-	-	-	-	-	-	-	-	-
Q	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-
R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-
S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-
T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-
U	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-
V	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-
W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-
X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	8	-	-
Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-
Z	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+

TABLE IV
CONFUSION MATRIX FOR RUSSIAN LETTERS RECOGNITION

	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ы	Э	Ю	Я
А	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Б	-	8	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
В	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Г	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Д	-	-	-	-	9	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Е	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ё	-	-	-	-	-	3	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3
Ж	-	-	-	-	-	-	-	4	6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
З	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
И	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Й	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
К	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Л	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
М	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Н	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
О	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
П	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	-	-	-	-	-	-	-	-	-	-	-	4	-	-
Р	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	6	-	-
С	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-
Т	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	-	-	-	-	-	-	-	-	4	-	-
У	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-
Ф	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-
Х	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9	-	-	-	-	-	-	-	-
Ц	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-
Ч	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-
Ш	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-
Щ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-
Ы	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-
Э	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-
Ю	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-
Я	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+

VII. FINGERSPELLING SYNTHESIS

Fingerspelling synthesis system creates 3D animation of the upper half of a human figure [26]. The system uses 3D articulatory model approximating the top surface of the figure by polygonal meshes. The meshes are divided into body segments such as arms, forearms, palm, knuckle-bones, face, internal mouth, etc. The animation model allows expressing both manual and non-manual components of sign language. The manual component is fully represented by rotations of joint connections. The

joints connections are composed in a tree structured hierarchy and every joint is assigned at least to one body segment. Thus the rotation of one segment causes rotations of other segments with lower hierarchy. Joint limits prevent non-anatomic positions or poses of the animation model.

The animation of the non-manual component employs the joint connections as well as moving of control points [26] and morph targets [27]. The joint connections ensure movements of shoulders, neck, skull, eyeballs and jaw. In contrast, the control points and morph targets allow us to change the local shape of the face, lips, or tongue. The control points have fixed positions on the polygonal mesh and their translation in 3D causes local deformations in face and tongue. Contrarily the morph targets are one or combination of more manually remodeled 3D positions of vertices of polygonal meshes. A complex gesture is expressed as weighted combination of these morph targets at a time [28].

Animation model is controlled via animation frames that are composed into animation trajectories. One animation trajectory describes the time sequence of values controlling particular joint connection, control point or weight of morph target. The animation frame does not store directly the values for shoulder, elbow and wrist joints (7 DOF), but includes pose matrix P_{4x4}. Two P matrices determine the locations of the wrist, the direction and twist of the palms separately for both arms. For P in particular animation frame the inverse kinematics module (IK) determines the final pose of the arms.

Target languages for fingerspelling synthesis are Czech and English. Specifically, we consider 26 letters and 10 numerals for American Sign Language (ASL) and 46 letters and 10 numerals for Czech Sign Language (CSL). To get descriptions of these signs, we used SignEditor [6], see Fig. 15. SignEditor allows the designer of the system to create and verify a set of signs (lexicon). In the first step a new sign must be manually entered in a notation system. SignEditor uses the Hamburg notation system (HamNoSys) [29]. This notation determines a set of rules to capture the sign in the string of symbols. The feedback of SignEditor is both the animation model and the control module transferring notation into animated trajectories. In the second step the control module will automatically create animation trajectories. The designer of the system can immediately verify the result of the transfer of the new sign to the animation.

A new feature of SignEditor is the export of entered signs directly to the animation trajectories. Thus pre-processed signs are directly loaded into fingerspelling synthesis during startup and they do not need to be transferred from their symbolic strings repeatedly. The fingerspelling synthesis system concatenates loaded trajectories to one continuous trajectory according to the entered utterance in real time. Piecewise linear interpolation is used to get fluent transition between two neighbor signs.

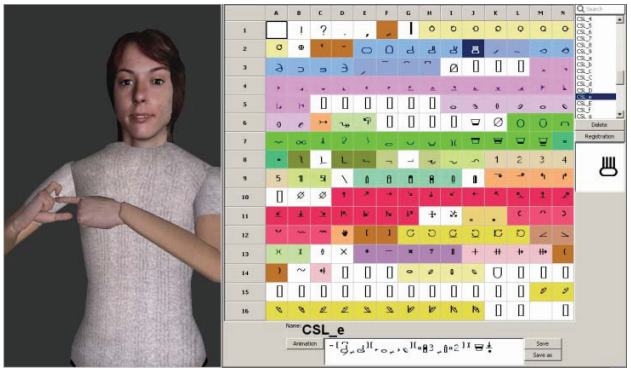


Fig. 15. The screen shot of SignEditor. On left the avatar, on right table of symbols separated to groups by colors.

The duration of the transition (the number of animation frames) is evaluated from animation frames of concatenated signs on-line to get natural transition [30].

CSL allows using both one- and two-handed fingerspelling alphabet. We chose the two-handed variant. Generally, hand shapes of two-handed signs are simpler. For two-handed variant, several CSL letters use only the dominant hand (seven letters). Others are always expressed with both hands located in front of the body. CSL numerals 0-5 are one-handed and 6-9 are two-handed. The dominant hand forms the base shape of the letter and the non-dominant hand has the same or simpler shape and is in contact with the dominant hand. HamNoSys has a rich repertoire for these contacts and they can be successfully converted to the avatar animation [31]. Some letters in CSL also include a simple movement of arms. This movement allows expressing the diacritic of some CSL letters.

Different situation arises for letters, and numerals in ASL. Expression of these signs is by one hand and requires very complex shapes of the dominant hand. Current state of the control module does not allow automatic conversion all ASL signs. For example, letter E includes multiple contacts between index finger, ring finger, middle finger and the thumb, letter D and numerals 6-9 incorporate a touch of thumb on the index finger as well as on other fingers. M and N letters include an intersection of thumb between remaining fingers and the letter R use crossing fingers. On the other hand, a precise animation of these signs is very important because for example crossing fingers distinguishes letters R and U. Hence the considered ASL signs must be manually specified. For this purpose we have extended SignEditor allowing direct editing of all joint connections of the dominant hand and saving corrected animation frames.

Animation frames are generated with a fixed frame rate. Therefore, they directly determine the speed of the animation. Since speed of fingerspelling for different sign languages differs, the number of animation frames produced by SignEditor must be checked to get the natural rate of resulting animation.

The entry of the fingerspelling synthesis system is an

utterance expressed in letters of the target language which is automatically transferred into 3D animation in real time. Firstly, the synthesis system finds signs for whole words of the utterance. Thus digits and isolated letters separated by spaces are directly chosen from the lexicon. Other words, abbreviations etc. have to be spelled. Because a clear separation of a spelled word from other signs is needed, a special "space" sign is inserted at the beginning and the end of each such word. This special sign puts down avatar's hands and the letters within words are directly connected without an interruption.

VIII. Speech Synthesis

Two TTS systems are applied in our global system: Open MARY TTS [32] for the English and Turkish languages developed by DFKI (Germany); Russian TTS engine developed by UIIP (Belarus) and SPIIRAS (Russia) [33]. Unit selection speech synthesis method is used for English, HMM-based speech synthesis method is applied for Turkish, and compilative allophone-diphone based synthesis method was realized for Russian. Male and female voices are available for English and Russian and there are only male voices for Turkish. TTS was realized as a web-based service, which waits for messages from the web-server.

IX. Conclusion

We have developed a real time system that performs fingerspelling to speech and speech to fingerspelling in different languages. The demo videos of the system can be found in [34][35][36].

X. ACKNOWLEDGEMENTS

This research was supported by the following grants and projects: Ministry of Education of the Czech Republic, project No. ME08106; Grant Agency of the Czech Republic, project No. GACR 102/09/P609; University of West Bohemia, project No. SGS-2010-054; RFBR and TÜBİTAK in the framework of the bilateral Russian-Turkish project (# 09-07-91220 / 108E113), Grant of the President of Russia (# MK-64898.2010.8); Turkish State Planning Organization (DPT) TAM Project, number 2007K120610 and TÜBİTAK BİDEB 2211; EU FP7 Marie Curie IEF NOVICOM.

References

[1] Liwicki, S. and Everingham, M., Automatic recognition of fingerspelled words in British Sign Language. In: Proceedings of CVPR4HB'09. 2nd IEEE Workshop on CVPR for Human Communicative Behavior Analysis, Thursday June 25th, Miami, Florida., pp. 50-57, 2009.

[2] P. Goh and E.-J. Holden, Dynamic fingerspelling recognition using geometric and motion features, in IEEE International Conference on Image Processing, pp. 2741 – 2744, Atlanta, GA USA, 2006.

[3] Gui, L . Thiran, J.P. and Paragios, N. Finger-spelling Recognition within a Collaborative Segmentation/Behavior Inference Framework. In Proceedings of the 16th European Signal Processing Conference (EUSIPCO-2008), Switzerland, 2008

[4] R. Elliott, J.R.W. Glauert, J.R. Kennaway, and I. Marshall. The development of language support for the visicast project. In 4th Int. ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2000). Washington DC.Assistive Technologies. Washington DC, 2000.

[5] Kennaway JR. Synthetic animation of deaf signing gestures. 4th International Workshop on Gesture and Sign Language in Human-Computer Interaction. Springer-Verlag, LNAI vol 2298, pp 146-157, 2001

[6] Zdeněk Krňoul, Jakub Kanis, Miloš Železný, and Luděk Müller. Czech text-to-sign speech synthesizer. Machine Learning for Multimodal Interaction, Series Lecture Notes in Computer Science, 4892:180–191, 2008.

[7] Rabiner L., Juang B. Speech Recognition, Chapter in Springer Handbook of Speech Processing (Benesty, Jacob; Sondhi, M. M.; Huang, Yiteng, eds.), NY: Springer, 2008.

[8] Young S. et al. The HTK book version 3.4 Manual. Cambridge University Engineering Department, Cambridge, UK, 2006

[9] Dutoit T., Bozkurt B. Speech Synthesis, Chapter in Handbook of Signal Processing Acoustics, D. Havelock, S. Kuwano, M. Vorländer, eds. NY: Springer. Vol 1, pp. 557-585, 2009.

[10] Ebru Arisoy, Dogan Can, Siddika Parlak, Hasim Sak and Murat Saraclar, "Turkish Broadcast News Transcription and Retrieval," IEEE Transactions on Audio, Speech, and Language Processing, 17(5):874-883, July 2009

[11] Allen J. G., Xu R. Y. D. and Jin J. S., Object tracking using camshift algorithm and multiple quantized feature spaces, in Proceedings of the Pan-Sydney area workshop on Visual information processing, ACM International Conference Proceeding Series Vol. 100 Australian Computer Society, Inc., Darlinghurst, Australia, 2004

[12] Exner, D., Bruns, E. , Kurz, D., Grundhöfer, A., Bimber, O., Fast and Reliable CAMShift Tracking, PhD thesis Bauhaus- Universitat Wiemar, Germany 2009.

[13] Ojala, T.,Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions , Pattern Recognition, vol. 29, 51-59, 1996.

[14] Nguyen, T. T.; Binh, N. D. & Bischof, H., An active boosting-based learning framework for real-time hand detection., in 'FG' , IEEE, pp. 1-6, 2008

[15] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 24, 971-987, 2002.

[16] Hu, M.K, Visual pattern recognition by moment invariants, IEEE Transactions on Information Theory 8 , pp. 179–187 1962.

[17] Lin C, Hwang C. New forms of shape invariants from elliptic Fourier descriptors. Pattern Recognition, 20(5): 535–45, 1987.

[18] Kuhl FP, Giardina CR. Elliptic Fourier features of a closed contour. Computer Graphics & Image Processing, 18:236–58, 1982.

[19] Yoruk E., Konukoglu E., Sankur B. and Darbon J.Shape-Based Hand Recognition IEEETransactions on Image Processing, Vol. 15, No. 7, July 2006.

[20] Liu R., Li Z., Jia J. Image partial blur detection and classification. In CVPR, 2008.

[21] Levenshtein V.I. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10: 707-10. 1966

[22] Sak, H., Saraçlar, M. and Güngör, T., On-the-fly Lattice Rescoring for Real-time Automatic Speech Recognition, Interspeech, Makuhari, Japan, September 2010.

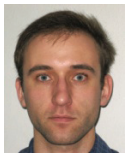
[23] Mohri M., Pereira F., and Riley M. Weighted finite-state trans-ducers in speech recognition. Computer Speech & Language, vol. 16, no. 1, pp. 69–88, 2002.

[24] Sak, H., Güngör, T., and Saraçlar, M., Resources for Turkish morphological processing, Language Resources and Evaluation. Springer, 2010.

[25] Rabiner L., Juang. Fundamentals of Speech Recognition New Jersey: Prentice-Hall, Englewood Cliffs, 1993.

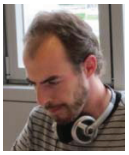
[26] Krňoul, Z. and Železný, M. Realistic face animation for a Czech Talking Head . Lecture Notes in Artificial Intelligence, Lecture notes in artificial intelligence, no. 3206, 3206, p. 603-610, Springer, Berlin, 2004.

[27] Zdeněk Krňoul. New features in synthesis of sign language addressing non-manual component. 4th Workshop on Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, ELRA, 2010.
[28] Parke F. I., Waters K. Computer Facial Animation, A K Peters, Ltd, Wellesley, MA 02482, 2 edition, 2008.
[29] http://www.sign-lang.uni-hamburg.de/projects/ hamnosys.html
[30] Krňoul, Z. and Železný, M. : Translation and Conversion for Czech Sign Speech Synthesis . Lecture Notes in Artificial Intelligence, 4629, p. 524-531, 2007.
[31] Kanis Jakub and Krňoul Zdeněk : Interactive HamNoSys Notation Editor for Signed Speech Annotation . p. 88-93, ELRA, 2008.
[32] http://mary.dfki.de
[33] Hoffmann R., Jokisch O., Lobanov B., Tsirulnik L., Shpilewsky E., Piurkowska B., Ronzhin A., Karpov A. Slavonic TTS and STT Conversion for Let's Fly Dialogue System, In Proceedings of the 12-th International Conference on Speech and Computer SPECOM, Moscow, Russia, 2007, pp. 729-733.
[34] http://www.cmpe.boun.edu.tr/pilab/pilabfiles/demos/enterface2010/Conversation.m4v
[35] http://www.cmpe.boun.edu.tr/pilab/pilabfiles/demos/enterface2010/Avatar.mp4
[36] http://www.cmpe.boun.edu.tr/pilab/pilabfiles/demos/enterface2010/Integrated_System.wmv



Pavel Campr graduated in cybernetics from the University of West Bohemia (UWB) in 2005. As a Ph.D. candidate at the Department of Cybernetics, UWB, his research interests focus on gesture and sign language recognition, computer vision, machine learning and multimodal human-computer interaction. He is participating in the research projects MUSSLAP (Multimodal Human Speech and Sign Language Processing for Human-Machine Communication), ARET (Automatic Reading of Educational Texts for Vision Impaired Students) and POJABR (Language handicap elimination for hearing-impaired students via automatic language processing). He is also teaching assistant and maintainer of the departmental website.

Erinç Dikici received his B.S. degree in Telecommunication Engineering from Istanbul Technical University (ITU), Turkey, in 2006, and his M.S. degree in Electrical and Electronics Engineering from Bogaziçi University in 2009. He is currently a Ph.D. student at the same department and a research asistant at BUSIM: Bogaziçi University Center for Signal and Image Processing. His research interests include speech and audio processing, specifically, speaker and speech recognition, and music information retrieval.

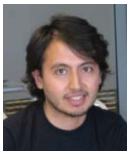


Marek Hruz was born in 1983 in Slovakia. He received his M.S. degree in cybernetics from the University of West Bohemia (UWB) in 2006. As a Ph.D. candidate at the Department of Cybernetics, UWB, his research interests focus on hand gesture and sign language recognition, particularly tracking and image parametrization, computer vision, machine learning and multimodal human-computer interaction. He is participating in the research projects MUSSLAP and POJABR and is a teaching assistant at the Department of Cybernetics, UWB.

Alp Ahmet Kindiroglu graduated from Computer Science and Engineering at Sabanci University,Turkey, in June 2008. Since then he has been working as a Tubitak funded research assistant in the Perceptual Intelligence Laboratory of Bogazici University, working towards a MSc degree in Computer Engineering. His main research interests are hand gesture recognition, sign language recognition and human computer interactions.

Zdenek Krnoul was born in Czech Republic in 1979. He received the PhD degree in Artificial intelligence from University of West Bohemia in Pilsen, in 2008. Since 2009 he is postdoctoral researcher at the Department of Cybernetics, University of West Bohemia in Pilsen. He also works as an application developer in the projects POJABR and ARET. He received "Golden Lips Award for Intelligibility" on "Visual Speech Synthesis Challenge" at the conference INTERSPEECH 2008, Brisbane. The current research interests include visual speech and sign language synthesis, mainly coarticulation of visual speech, methods for lip tracking and detection, combination of the non-manual component and the manual component of sign speech and 3D modeling and animation of human face and body.

Alexander Ronzhin received the M.Sc. diploma from St. Petersburg State University of Airspace Instrumentation in 2010. His main research interests include processing audio-visual signals for intelligent spaces, text-to-speech systems, user localization, tracking and video surveillance in the smart room, audio-visual speech analysis and synthesis. Currently he is a programmer of Speech and Multimodal Interfaces Laboratory of SPIIRAS. He is the (co)author of 7 papers in proceedings of international conferences.

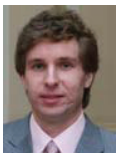


Hasim Sak received his B.S. degree in 2000 from the Computer Engineering Department at Bilkent University, and M.S. degree in 2004 from the Computer Engineering department at Boğaziçi University, where he is now pursuing his Ph.D. degree. On his thesis, he is focusing on the language modeling and speech decoding challenges associated with agglutinative languages and rich morphology. From 2000 to 2005, he worked in a company developing speech technologies for Turkish. His main research interests include speech recognition, speech synthesis, statistical language modeling, morphological parsing, spelling correction, and morphological disambiguation.

Daniel Schorno (Composer) was born in Zurich, Switzerland, studied composition, cello and conducting in London with Melanie Daiken, William Mival and Lawrance Leonard, and electronic and computer music in The Hague/Netherlands, with Clarence Barlow and Joel Ryan. Invited by Michel Waisvisz he lead STEIM - the renown Dutch Studio for Electro Instrumental Music, and home of 'New Instruments' - as Artistic Director from 2001-05 and holds currently the position as Creative Director. He has collaborated with musicians, artists, choreographers, ensembles and organisations like Frances-Marie Uitti, Ernest Rombaut, Daniel Koppelman, Netochka Nezvanova, Laetitia Sonami, Francisco Lopez, Anne Laberge, Steina Vasulka, Frank van de Ven, José Navas, Pascal Boudreault, the Dutch 'Nieuw' and Insomnio Ensembles, and organisations like the Forum Neues Musiktheater Stuttgart and the Theremin Institute in Moscow.

Lale Akarun received the BS and MS degrees in Electrical Engineering from Bogazici University, Istanbul, Turkey, in 1984 and 1986, respectively, and the PhD degree from Polytechnic University, Brooklyn, New York, in 1992. From 1993 to 1995, she was Assistant Professor of Electrical Engineering at Bogazici University, where she is now Professor of Computer Engineering. Her current research interests are in image processing, computer vision, and computer graphics.

Oya Aran received the PhD degree in Computer Engineering from Bogazici University, Istanbul, Turkey in 2008. During her PhD study she focused on hand gesture recognition and sign language recognition. She has published papers in leading computer vision and pattern recognition journals and conferences. She was awarded a Marie Curie Intra-European fellowship in 2009 with NOVICOM (Automatic Analysis of Group Conversations via Visual Cues in Non-Verbal Communication) project. She joined the Idiap Research Institute, Martigny, Switzerland in June 2009 as a postdoctoral researcher and a Marie Curie fellow. Her research interests include pattern recognition, machine learning, computer vision, human–computer interaction and social computing.



Alexey Karpov received the M.Sc. diploma from St. Petersburg State University of Airspace Instrumentation and Ph.D. degree in computer science from St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), in 2002 and 2007, respectively. His main research interests include automatic speech recognition, text-to-speech systems, multimodal interfaces based on speech and gestures, audio-visual speech analysis and synthesis. Currently he is a senior researcher of Speech and Multimodal Interfaces Laboratory of SPIIRAS. He has been the (co)author of more than 80 papers in refereed journals and proceedings of international conferences. Dr. Karpov is the co-chairman of the organizing committee of series of International conferences on Speech and Computer SPECOM, as well as the current member of ISCA, EURASIP (Local Liaison Officer), IAPR and OpenInterface associations.

Murat Saraclar received the B.S degree from the Electrical and Electronics Engineering Department, Bilkent University, Ankara, Turkey, in 1994 and the M.S.E. and Ph.D. degrees from the Electrical and Computer Engineering Department, The Johns Hopkins University, Baltimore, MD, in 1997 and 2001, respectively. He is currently an associate professor in the Electrical and Electronic Engineering Department, Boğaziçi University, Istanbul, Turkey. From 2000 to 2005, he was with the Multimedia Services Department, AT&T Labs—Research. His main research interests include all aspects of speech recognition, its applications, as well as related fields such as speech and language processing, human–computer interaction, and machine learning. Dr. Saraçlar is currently a member of the IEEE Signal Processing Society Speech and Language Technical Committee.

Milos Zelezny was born in Plzen, Czech Republic, in 1971. He received his Ing. (=M.S.) and Ph.D. degrees in Cybernetics from the University of West Bohemia, Plzen, Czech Republic (UWB) in 1994 and in 2002 respectively. He is currently a lecturer at the UWB. He has been delivering lectures on Digital Image Processing, Structural Pattern Recognition and Remote Sensing since 1996 at UWB. He is working in projects on multi-modal speech interfaces (audio-visual speech, gestures, emotions, sign language). He is a member of ISCA, AVISA, and CPRS societies. He is a reviewer of the INTERSPEECH conference series.

ISBN 978-9-057-76213-0

UNIVERSITEIT VAN AMSTERDAM